



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

THESIS

**STANDARDIZING METHODS FOR WEAPONS ACCURACY
AND EFFECTIVENESS EVALUATION**

by

Nicholas D. Levine

June 2014

Thesis Advisor:
Second Reader:

Morris Driels
Christopher Adams

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2014	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE STANDARDIZING METHODS FOR WEAPONS ACCURACY AND EFFECTIVENESS EVALUATION			5. FUNDING NUMBERS	
6. AUTHOR(S) Nicholas D. Levine				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number N/A .				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) The Joint Technical Coordinating Group for Munitions Effectiveness desires a standardized toolbox and methodologies for evaluating weapons accuracy. Using statistical distributions, a method is presented for both unguided and Global Positioning System-guided munitions. The statistics used to describe a sample of weapons firings will not only describe the weapons' accuracy, but will also be utilized by the Joint Weaponing System to calculate the weapons' effectiveness against specified targets. Since the precision of the inputs and statistics used to describe the accuracy of the weapons is sensitive, it is imperative that the inputs are accurately modeled as they can lead to drastically different effectiveness results. Analysts must also carefully consider the assumptions used in the application of specific statistical distributions. The toolbox and methods presented here illustrate the differences among techniques and the pros and cons of each.				
14. SUBJECT TERMS weaponeering, error probable, delivery accuracy, effectiveness, GPS, guided			15. NUMBER OF PAGES 127	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**STANDARDIZING METHODS FOR WEAPONS ACCURACY AND
EFFECTIVENESS EVALUATION**

Nicholas D. Levine
Lieutenant Commander, United States Navy
B.S., University of Denver, 2003
M.E.M., University of Denver, 2009
M.S., University of Colorado, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
June 2014**

Author: Nicholas D. Levine

Approved by: Morris Driels
Thesis Advisor

Christopher Adams
Second Reader

Knox Millsaps
Chair, Department of Mechanical and
Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Joint Technical Coordinating Group for Munitions Effectiveness desires a standardized toolbox and methodologies for evaluating weapons accuracy. Using statistical distributions, a method is presented for both unguided and Global Positioning System-guided munitions. The statistics used to describe a sample of weapons firings will not only describe the weapons' accuracy, but will also be utilized by the Joint Weaponeering System to calculate the weapons' effectiveness against specified targets. Since the precision of the inputs and statistics used to describe the accuracy of the weapons is sensitive, it is imperative that the inputs are accurately modeled as they can lead to drastically different effectiveness results. Analysts must also carefully consider the assumptions used in the application of specific statistical distributions. The toolbox and methods presented here illustrate the differences among techniques and the pros and cons of each.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION AND OBJECTIVES.....	1
II.	STATISTICS BACKGROUND.....	3
	A. SAMPLES, MEAN, AND VARIANCE.....	3
	B. UNIVARIATE NORMAL DISTRIBUTION.....	5
	C. BIVARIATE NORMAL DISTRIBUTION.....	7
	D. CIRCULAR NORMAL DISTRIBUTION.....	8
	E. RAYLEIGH DISTRIBUTION.....	9
	F. PROBABILITIES.....	11
III.	WEAPONNEERING BACKGROUND.....	13
	A. DEFINITIONS.....	13
	B. ERROR PROBABLE.....	14
	C. MATHEMATICAL RELATIONSHIPS OF REP, DEP, AND CEP.....	16
	D. ACCURACY MODELS.....	17
	E. KOLMOGOROV-SMIRNOV (K-S) TEST.....	20
	F. LEGACY METHODS.....	20
IV.	TOOLBOX DEVELOPMENT.....	23
	A. GROSS ERROR EXTRACTION FOR RANGE AND DEFLECTION.....	24
	B. DOUBLE NORMAL DISTRIBUTION.....	25
	C. DOUBLE RAYLEIGH DISTRIBUTION.....	26
	D. SINGLE NORMAL DISTRIBUTION.....	27
	E. SINGLE RAYLEIGH DISTRIBUTION.....	28
	F. MATLAB'S COMPILER DEPLOYMENT TOOL.....	28
V.	TOOLBOX IMPLEMENTATION FOR ACCURACY ASSESSMENT.....	31
	A. DATA IDENTIFICATION.....	31
	B. RANGE AND DEFLECTION DATA.....	31
	1. GE_extractXY.....	31
	2. SN_CDF_data.....	32
	3. DN_CDF_data.....	33
	C. RADIAL DATA.....	34
	1. SR_CDF_data.....	34
	2. DR_CDF_data.....	34
	D. ALTERNATIVE APPROACH.....	36
VI.	CALCULATING WEAPON EFFECTIVENESS.....	37
	A. PROBABILITY OF KILL CALCULATIONS.....	37
	B. MONTE CARLO APPROACH.....	37
	C. EXPECTED VALUE THEOREM.....	38
	D. P_{HIT}/P_{NM} METHODOLOGY.....	39
	E. THE JMEM WEAPONNEERING SYSTEM.....	39
VII.	IMPLEMENTATION FOR EFFECTIVENESS.....	43
	A. RANGE AND DEFLECTION DATA.....	43

	B.	RADIAL DATA.....	44
VIII.		COMPARISON OF DATA ANALYSIS METHODS AND THEIR EFFECT ON PK.....	47
	A.	PROBABILITY OF KILL.....	47
	B.	MEDIAN RADIAL ANALYSIS.....	49
	C.	ANALYSIS IN RANGE AND DEFLECTION.....	51
	D.	THE TOOLBOX-DOUBLE NORMAL ANALYSIS.....	52
IX.		FURTHER EXAMPLE TOOLBOX APPLICATIONS.....	57
	A.	CASE STUDY 1.....	57
		1. Calculating Delivery Accuracy.....	57
	B.	CASE STUDY 2.....	63
		1. Delivery Accuracy.....	63
		2. Current Methodology.....	67
	C.	CASE STUDY 3.....	69
		1. Delivery Accuracy.....	69
X.		THE TOOLBOX AND GPS GUIDED WEAPONS.....	71
	A.	THE TOOLBOX AND GPS GUIDED DATA.....	71
	B.	GPS BACKGROUND.....	72
	C.	NAV ERROR.....	75
	D.	RESOLUTION.....	76
	E.	GPS GUIDED WEAPON CASE STUDY.....	77
		1. HDOP and VDOP.....	77
		2. HDOP/VDOP and the Test Data.....	78
		3. Data Refinement.....	80
		4. Modeling Impact Angle.....	80
		5. Updated Model.....	81
XI.		CONCLUSIONS AND RECOMMENDATIONS.....	85
		APPENDIX	87
	A.	MATLAB CODE - DN_CDF_DATA.....	87
		1. MATLAB CODE DN_myfit.....	89
	B.	MATLAB CODE - DR_CDF_DATA.....	89
		1. MATLAB CODE DR_myfit.....	91
	C.	MATLAB CODE - GE_EXTRACT_XY.....	92
	D.	MATLAB CODE - SN_CDF_DATA.....	95
	E.	MATLAB CODE - SR_CDF_DATA.....	96
	F.	MATLAB CODE - GE_EXTRACT.....	98
	G.	MATLAB CODE - PHIT/PNM.....	100
	H.	MATLAB CODE - GE_EXTRACT_GAUSSIAN.....	102
	I.	GPS WEAPON ACCURACY CALCULATOR.....	103
		LIST OF REFERENCES	107
		INITIAL DISTRIBUTION LIST	109

LIST OF FIGURES

Figure 1.	Distribution of miss distances.....	4
Figure 2.	Normal probability density function (from [1])...	5
Figure 3.	Probability area under a normal curve (from [1])	6
Figure 4.	Bivariate normal distribution (from [1]).....	8
Figure 5.	Rayleigh PDF.....	11
Figure 6.	Distribution of impact points (from [1]).....	14
Figure 7.	Definition of REP (from [1]).....	15
Figure 8.	Definition of DEP (from [1]).....	15
Figure 9.	Definition of CEP (from [1]).....	16
Figure 10.	Normal fit to test data.....	18
Figure 11.	Double Normal fit to test data.....	19
Figure 12.	P_{HIT}/P_{NM} Methodology (from [1])	21
Figure 13.	The Toolbox.....	23
Figure 14.	GE_extractXY Input/Output.....	25
Figure 15.	DN_CDF_data Input/Output.....	26
Figure 16.	DR_CDF_data Input/Output.....	27
Figure 17.	SN_CDF_data Input/Output.....	27
Figure 18.	SR_CDF_data Input/Output.....	28
Figure 19.	Single Distribution flow chart.....	35
Figure 20.	Effectiveness flowchart.....	45
Figure 21.	Test data impact points.....	49
Figure 22.	Pk based on CEP.....	50
Figure 23.	Comparison: CEP vs. REP/DEP.....	52
Figure 24.	DN_CDF_data graph for range.....	53
Figure 25.	DN_CDF_data graph for deflection.....	54
Figure 26.	Comparison: DN method/SN method/Radial method...	55
Figure 27.	Range and deflection plot for Weapon A.....	59
Figure 28.	Weapon A - DN_CDF_data graph for range.....	60
Figure 29.	Weapon A - DN_CDF_data plot for deflection.....	61
Figure 30.	Weapon A - SN_CDF_data for range.....	62
Figure 31.	Weapon A - SN_CDF_data for deflection.....	63
Figure 32.	Weapon B - DN_CDF_data for range.....	65
Figure 33.	Weapon B - DN_CDF_data for deflection.....	66
Figure 34.	Weapon B - DR_CDF_data for radial misses.....	68
Figure 35.	Test data impact points.....	72
Figure 36.	GPS Weapon Accuracy Calculator.....	75
Figure 37.	HDOP plotted for 30 days.....	78
Figure 38.	VDOP plotted for 30 days.....	78
Figure 39.	Data vs. HDOP/VDOP (same impact angle).....	79
Figure 40.	Histogram of Impact Angles.....	81
Figure 41.	Data vs. HDOP/VDOP (generalized impact angle)...	82

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	DN_CDF_data results.....	53
Table 2.	Weapon A data.....	58
Table 3.	Weapon A - DN_CDF_data results.....	59
Table 4.	Weapon B data.....	64
Table 5.	Weapon B - DN_CDF_data results.....	64
Table 6.	Weapon B - DR_CDF_data results.....	67
Table 7.	Weapon C data.....	69
Table 8.	Weapon C - DN_CDF_data results.....	70
Table 9.	Weapon C - SN_CDF_data results.....	70
Table 10.	DN_CDF_data results.....	71
Table 11.	Standard GPS ranging errors.....	73

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CEP	circular error probable
CDF	cumulative density function
DEP	deflection error probably
DPI	desired point of impact
FAA	Federal Aviation Administration
G&C	guidance and control
GPS	Global Positioning System
GWDAP	GPS Weapons Delivery Accuracy Calculator
HDOP	horizontal dilution of precision
JMEM	Joint Munitions Effectiveness Manual
JTCG/ME	Joint Technical Coordinating Group for Munitions Effectiveness
JWS	Joint Weaponeering System
K-S	Kolmogorov-Smirnov test
MPI	mean point of impact
PDF	probability density function
P_{GE}	probability of gross error
P_{HIT}	probability of hit
P_{NM}	probability of near miss
REP	range error probable
TLE	target location error
UERE	user equivalent range error
VDOP	vertical dilution of precision

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Dr. Morris Driels for his continued support and guidance. I would also like to thank Dr. Jem Corcoran for teaching the statistical fundamentals required of me to complete this project.

To my wife, Jill, and kids, Dylan, Riley, Keegan and Sawyer, thank you for always giving me a reason to continue to work hard and excel. Your encouragement and smiles make the hard days so much easier.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION AND OBJECTIVES

The Joint Technical Group for Munitions Effectiveness (JTTCG/ME) receives weaponeering data from multiple services and representatives. Often the methodologies used to evaluate the data are different and not standardized. It is not that the methodologies employed are incorrect, but that the various procedures and descriptive statistics used lack consistency.

The objective of this thesis is to develop a standard process that can be utilized by all services and evaluators. In order to leverage computing power, MATLAB will be used to produce a toolbox that outputs the required parameters in a consistent manner. Upon implementation of the toolbox, the JTTCG/ME will have a set of processes and tools that create standardized solutions to weapons accuracy evaluations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. STATISTICS BACKGROUND

In evaluating the effectiveness and accuracy of weapons systems, we rely largely on statistical analysis to predict the accuracy of an upcoming weapons firing given the historical accuracy. These statistical values establish metrics for comparison among different systems and set the stage for evaluating which weapon is appropriate for a given scenario. The precision of this data is crucial as it will feed weapon effectiveness calculations, which are used to determine the probability of kill. This directly affects the number of weapons delivered to a target. The purpose of this chapter is to present the statistical concepts used throughout this thesis as they are applied to evaluate weapon accuracy.

A. SAMPLES, MEAN, AND VARIANCE

In all cases, we will evaluate independent samples of random occasions, shots or volleys, at a target. Suppose we are given a sample of 100 miss distances for an unguided bomb at a stationary target. We define the mean or average miss distance in Equation (1), where n is the sample size.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1}{100} (x_1 + x_2 + \dots + x_{100}) \quad (1)$$

Summing all the miss distances and dividing by the total number of observations results in the average miss distance. Additionally, it is useful to identify a quantity known as the sample variance, which is defined in Equation (2).

$$S_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{99} [(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_{100} - \bar{x})^2] \quad (2)$$

Although variance is not commonly utilized in practice, the square root of the variance is defined as the standard deviation (S_x) and is more widely utilized. The standard deviation is the measure of spread within the sample from the mean. To provide a visual example, Figure 1 represents a distribution of miss distances with $\bar{x} = 0$ and $S_x = 1.088$.

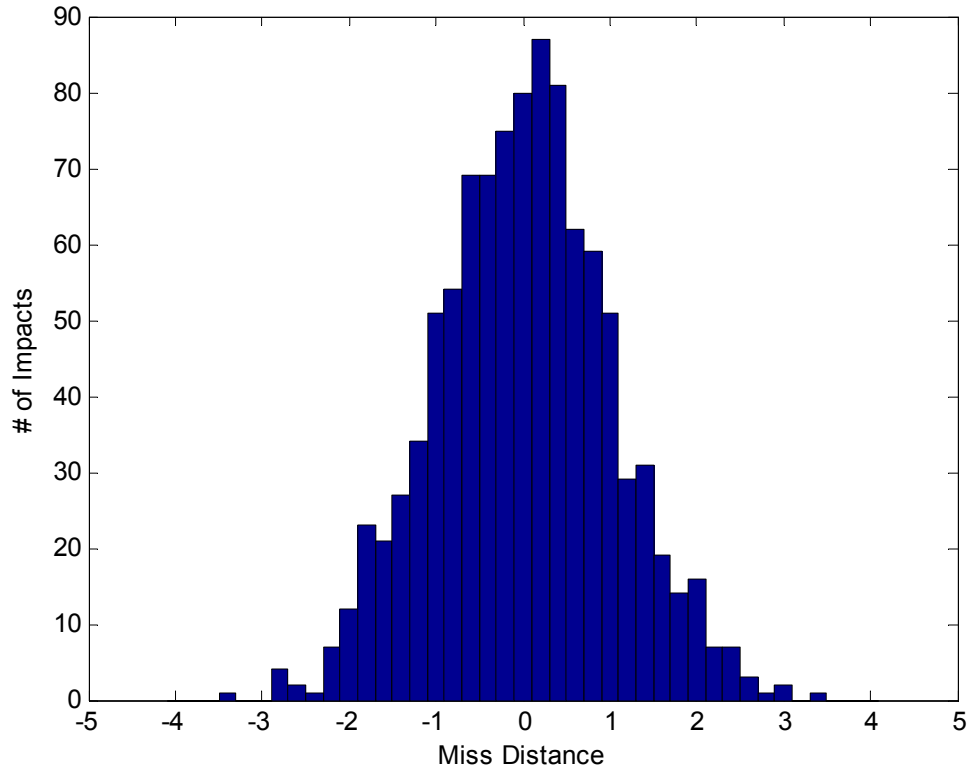


Figure 1. Distribution of miss distances

B. UNIVARIATE NORMAL DISTRIBUTION

In an effort to describe the data shown in Figure 1, it is desirable to generate a curve that represents the likelihood of an event or occurrence. If we assume that the miss distance is a continuous random variable, x , we can write the probability density function (PDF) for a normal distribution as defined in Equation (3).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-(x-\mu)^2/2\sigma^2\right] \quad (3)$$

Written in this manner, the symbols μ and σ represent the mean and standard deviation of the sample, respectively. These are more common symbolic representations of the statistics and will be used from here forward. A pictorial example of the normal PDF is shown in Figure 2. This depiction illustrates a sample with a slightly negative bias, or negative mean.

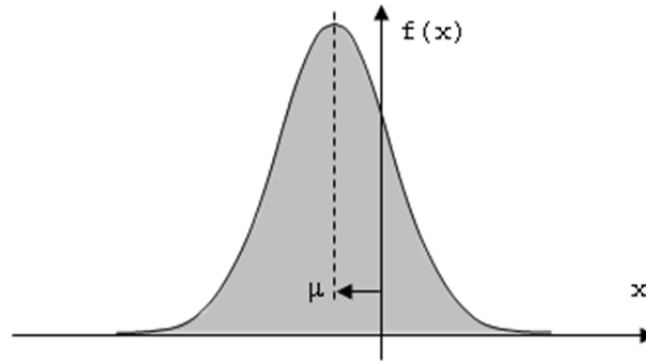


Figure 2. Normal probability density function (from [1])

This smooth, symmetrical, bell-shaped curve seemingly fits weapon accuracy data well [1]. The data in this case is characterized by a large number of hits, with the number of hits decreasing in frequency as you move away from the

target. The area under the PDF represents the probability of occurrence, and is pictorially shown in Figure 3.

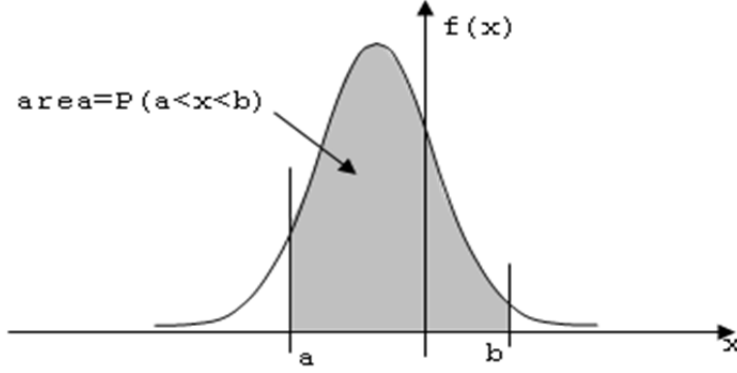


Figure 3. Probability area under a normal curve (from [1])

Thus, to find the probability of getting a miss distance between a and b , we integrate the PDF from a to b . Similarly, to find the probability that an event is less than or equal to X , we integrate the PDF from minus infinity to X . This integration results in the cumulative density function (CDF) and is written in Equation (4) for the normal distribution.

$$F(X) = P(x \leq X) = \int_{x=-\infty}^{x=X} f(x)dx = \int_{x=-\infty}^{x=X} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-(x-\mu)^2 / 2\sigma^2\right] dx \quad (4)$$

Integrating the CDF from minus infinity to infinity will result in a value of one (the probability of occurrence cannot be greater the one). The solution to Equation (4) is not trivial and is most often tabulated. A transformation to a standard normal distribution allows for use of the tabular values but will not be discussed here. See reference [2] for more information regarding this transformation.

C. BIVARIATE NORMAL DISTRIBUTION

In weaponeering applications it is common to assume that the miss distances in range and deflection are both independent and normally distributed [1]. Here, the miss distance in range is plotted on the vertical axis defined as the x-axis (short or long from the desired impact point) and the miss distance in deflection is plotted on the horizontal axis defined as the y-axis (left or right of the desired impact point). This is opposite of the standard mathematical convention. As the two distributions will not have the same values for mean and standard deviation, the samples can be represented by Equation (5); the bivariate normal PDF for independent and uncorrelated values of x and y.

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right] \quad (5)$$

As before, we can integrate the PDF, which results in the CDF for the bivariate normal distribution in Equation (6). In this equation, we are determining the probability of an event being less than both X and Y, respectively.

$$F(X, Y) = \int_{x=-\infty}^{x=X} \int_{y=-\infty}^{y=Y} \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right] dx dy \quad (6)$$

A pictorial example of the bivariate normal distribution is shown in Figure 4. It is not common, however, to evaluate data using the form in Equation (6). If the samples are independent and identically distributed, we are able to evaluate their statistics by using the univariate normal distribution and then exploiting convolutions in calculating probabilities that will be discussed later.

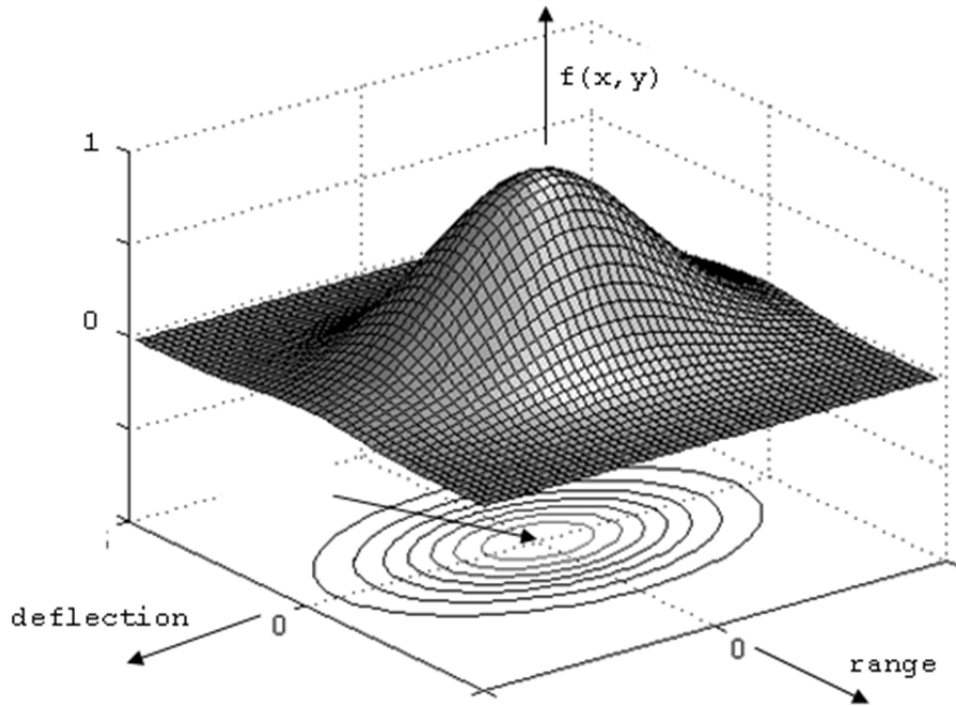


Figure 4. Bivariate normal distribution (from [1])

D. CIRCULAR NORMAL DISTRIBUTION

A circular normal distribution is a particular case of the bivariate normal distribution. In the case where the independent samples in range and deflection have means equal to zero and standard deviations that are equal in magnitude, the resulting distribution is called the circular normal distribution. The circular normal PDF is written in Equation (7).

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2}{2\sigma^2} + \frac{y^2}{2\sigma^2}\right] = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right] \quad (7)$$

In most cases, weaponeering miss distances in range and deflection will not meet the criteria to fit a circular normal distribution as the standard deviations in range and

deflection are rarely equal in magnitude. Rather than assume that the independent samples meet the criteria to be modeled by a circular normal distribution, we convert the data in range and deflection to radial miss distances, as in Equation (8).

$$r^2 = x^2 + y^2 \quad (8)$$

In order to emphasize the change in orientation, the x value corresponds to the miss distance in range and the y value corresponds to a miss distance in deflection. We will limit our discussion of the circular normal distribution here, as once the data is converted to radial miss distances its properties change.

E. RAYLEIGH DISTRIBUTION

The principles of the circular normal distribution are not easy to implement and it is uncommon that independent weapon firings will have miss distances in range and deflection that are identical univariate normal distributions. Rather than force the assumptions required by a circular normal distribution for statistical analysis, converting to radial miss distances results in much more convenient and accurate mathematical expressions. The resulting radial miss distances are not normally distributed. The resulting distribution is known as the Rayleigh distribution. The PDF and CDF for the Rayleigh distribution are shown in Equations (9) and (10).

$$f(r) = \frac{r}{\sigma^2} \exp\left[\frac{-r^2}{2\sigma^2}\right] \quad (9)$$

$$F(R)=1-\exp\left[\frac{-R^2}{2\sigma^2}\right] \quad (10)$$

It is imperative to illustrate a key difference in this distribution compared to those previously presented. The standard deviation in Equations (9) and (10) is not the standard deviation of the radial miss distances, but rather the common, or average, standard deviation of the range and deflection miss distances [1]. Usually, the standard deviation implemented in the Rayleigh distribution, denoted as σ_c , is written as in Equation (11).

$$\sigma_c = \frac{(\sigma_x + \sigma_y)}{2} \quad (11)$$

In the event that the data for range and deflection is unknown and the only data recorded is the radial miss distances, we can still find the common standard deviation. Exploiting statistical relationships between distributions, one can show that the common standard deviation is related to the standard deviation of the sample of radial miss distances by $\sigma_r = 0.655 * \sigma_c$, where σ_r is the standard deviation of the radial miss distances [1].

There are drawbacks to converting to radial miss distance as opposed to analyzing both range and deflection miss distances independently. Mainly, combining the standard deviations in range and deflection can significantly skew the data if the dispersions of each distribution are not close in magnitude. Consider artillery firings, in which it is common to have misses with a large dispersion in range and very little dispersion in deflection [3]. Converting to radial miss distances will

skew this fact and, unless range and deflection are evaluated separately, this fact will never be recovered. The positive in implementing the Rayleigh distribution is that unlike the univariate normal distribution, the value of the CDF for the Rayleigh distribution can be explicitly calculated and does not need tabular values to approximate the solution. A pictorial example of the Rayleigh distribution is provided Figure 5.

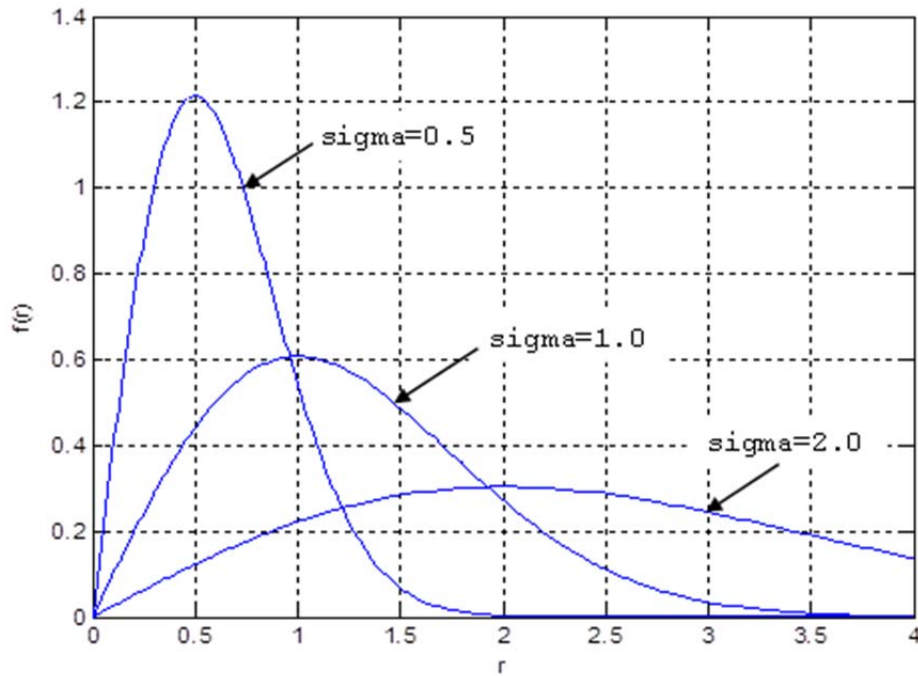


Figure 5. Rayleigh PDF

F. PROBABILITIES

The goal of fitting the data to the distributions described in this chapter is to calculate the probability of a miss for an occurrence. Ideally, the miss distance would always be zero and we would hit the target with every shot. In reality, this is unlikely. In order to exploit

some properties of probability, we must define independent events. We say that two events are independent of one another if the occurrence of one has no influence on the probability of the other [2]. For instance, the miss distance in range is assumed to be independent of the miss distance in deflection. Applying the special multiplication rule for independent events provided in Equation (12), we are able to evaluate range and deflection miss distances separately and still calculate a single probability of hitting the target.

$$P(A \text{ and } B) = P(A) \times P(B) \quad (12)$$

III. WEAPONEERING BACKGROUND

Although the statistics described in the previous chapter are useful when talking about the properties of the distributions, we must bridge the gap between the statistics and the terminology used by weaponeers. In this section, we will define terms utilized in weaponeering applications and show how the statistics relate to these terms.

A. DEFINITIONS

As mentioned previously, miss distances in the x axis direction are misses in range. This axis is aligned with the direction of travel on the weapon. Misses in the y axis direction are perpendicular to the direction of motion of the projectile and are in the deflection direction. The point at which we are aiming is called the desired point of impact (DPI). If we were to average all the miss distances in range and plot a horizontal line, this value would represent a range bias. Similarly, if we average all the miss distances in deflection and plot a vertical line, this would represent a deflection bias. The point where these two biases meet is called the mean point of impact (MPI) [1]. These concepts are pictorially represented in Figure 6.

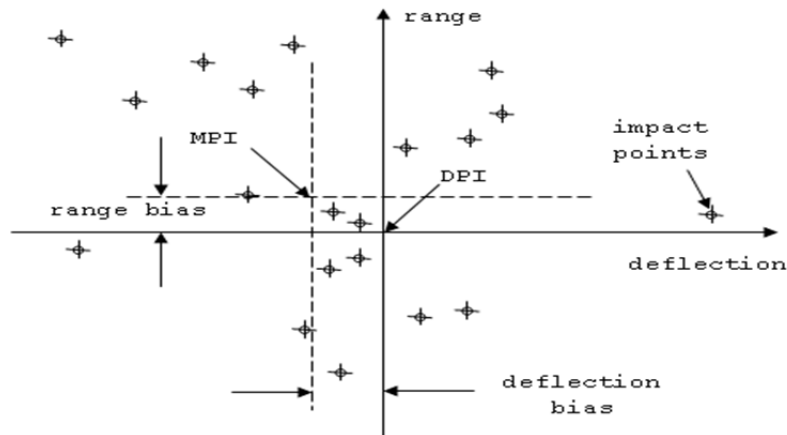


Figure 6. Distribution of impact points (from [1])

B. ERROR PROBABLE

Rather than discussing the specific statistics (mean or standard deviation) associated with the distribution of data, weaponeers use range error probable (REP) and deflection error probable (DEP). The REP is the distance from the DPI to a pair of lines perpendicular to the range axis such that fifty percent of the impact points in range lie between them. Similarly, the DEP is a pair of lines perpendicular to the deflection axis such that fifty percent of the impact points lies between them. Although REP and DEP categorize data in range and deflection, a more common error probable is the circular error probable (CEP).

CEP is defined as the radius of a circle from the desired impact point such that fifty percent of the impact points lie within the circle. Typically, REP, DEP and CEP are calculated after all biases are removed. Thus, the distributions are shifted so that their means are equal to zero [1]. Figures 7, 8, and 9 pictorially display REP, DEP

and CEP, respectively. Additionally, for the analysis in this thesis, we will evaluate data sets in which all biases are removed. Thus, the means of the distributions are usually set to zero.

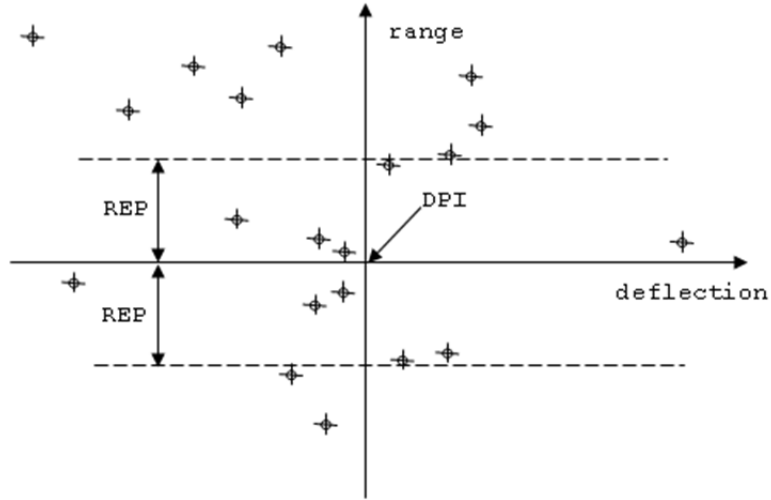


Figure 7. Definition of REP (from [1])

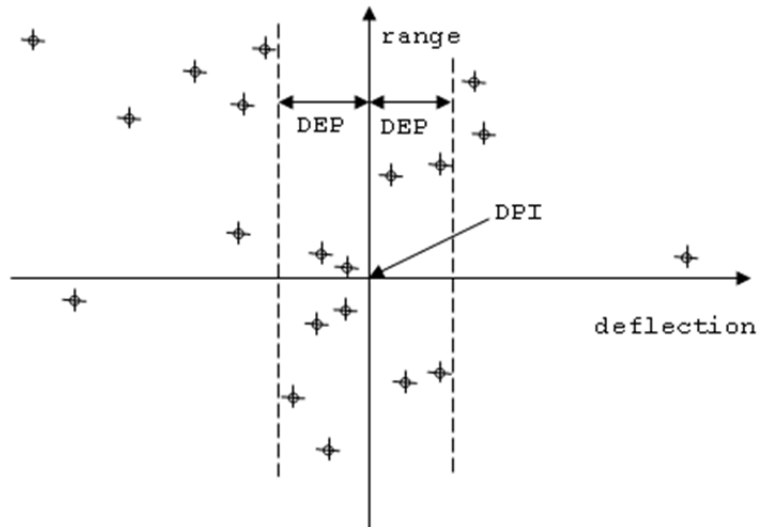


Figure 8. Definition of DEP (from [1])

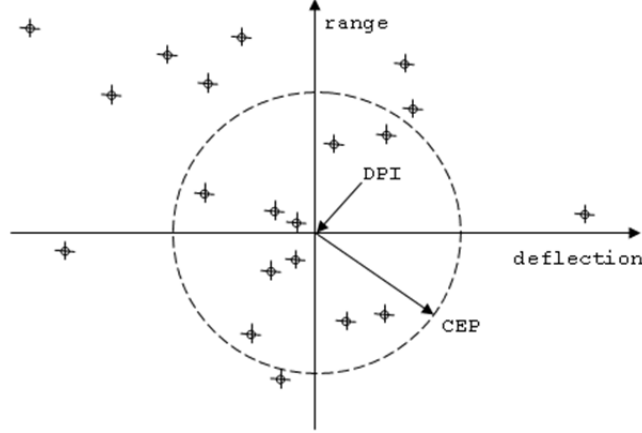


Figure 9. Definition of CEP (from [1])

C. MATHEMATICAL RELATIONSHIPS OF REP, DEP, AND CEP

Now that we have defined REP, DEP, and CEP, we will show how they are related to the standard deviation of the sample. First, we must assume the data is normally distribution with a mean of zero or that the bias has been removed. We must also assume that the data is independent in range and deflection. Given these assumptions, REP and DEP can be calculated using Equations (13) and (14).

$$REP = 0.6745\sigma_x \quad (13)$$

$$DEP = 0.6745\sigma_y \quad (14)$$

Similarly, for a Rayleigh distribution of radial miss distances, we can find the CEP is calculated implementing Equation (15).

$$CEP = 1.1774\sigma \quad (15)$$

Finally, if we are only given a value for CEP, we will assume that REP and DEP are equal as given by the common standard deviation. Thus, REP, DEP and CEP are related as shown in Equation (16).

$$REP = DEP = 0.573 \times CEP \quad (16)$$

See [1] for the full derivation of Equations (13), (14), and (15). The goal in finding REP and DEP, or CEP, is to feed these values into the calculations for weapons effectiveness or probability of kill. These calculations are both a function of weapon accuracy, as measured by the error probable values, and the lethal area of the weapon. Although we leave the lethal area of specific weapons to designers, given a lethal area our goal is to measure REP, DEP, and CEP with the highest precision possible in order to calculate the most accurate probability of kill.

D. ACCURACY MODELS

Thus far, we have assumed that the random test data can be accurately modeled by a normal or radial distribution. If we calculate an error probable based on a poorly fitting distribution, we cannot be confident that it truly describes the data with any accuracy. To provide an example wherein our assumptions can lead us astray, consider Figure 10. We have plotted the PDF and CDF for test data and assumed it to be normally distributed. Clearly, the statistical model does not fit the data with any precision and the model fails the Kolmogorov-Smirnov (K-S) test for goodness of fit at the sixty percent confidence level.

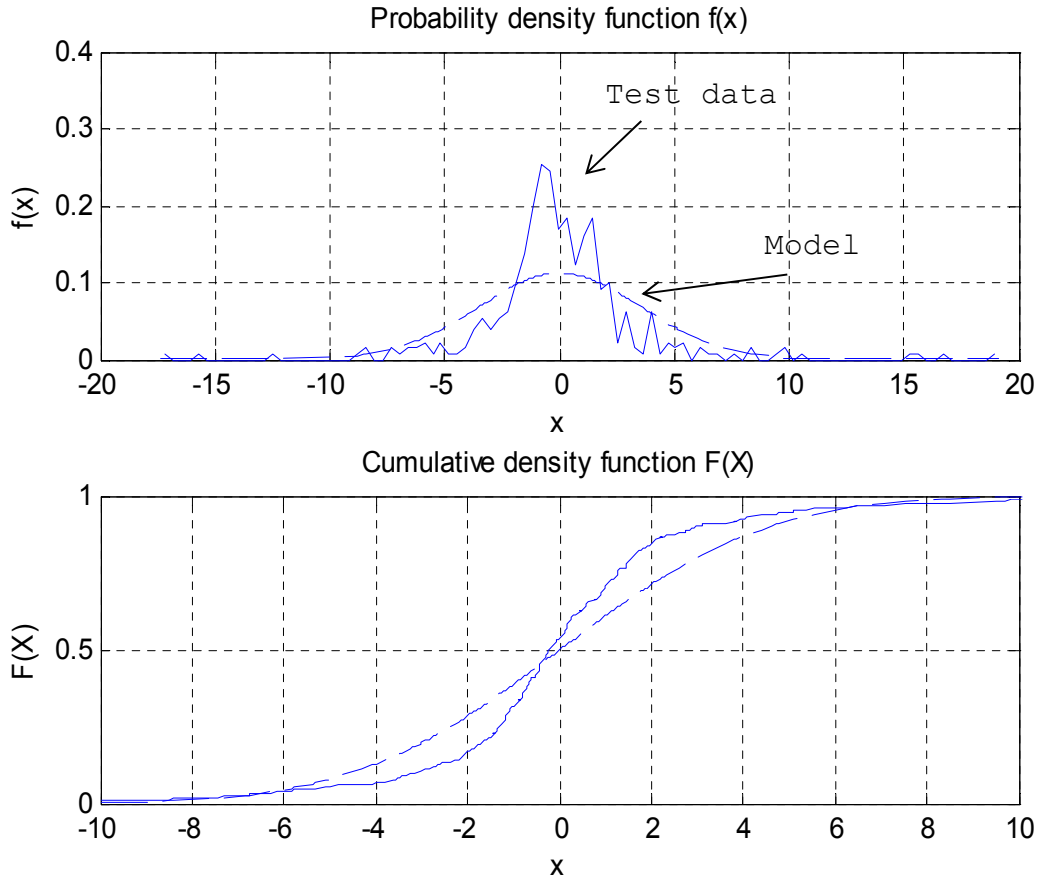


Figure 10. Normal fit to test data

It has been noted that some test data is better fitted to a linear combination of distributions in the linear or radial directions [4]. For the purposes of our analysis, this linear combination can take two forms. First, a linear combination of the normal PDF will be called a double normal PDF, and is shown in Equation (17).

$$f(x) = p \times \frac{1}{\sigma_1 \sqrt{2\pi}} \exp \left[\frac{-(x - \mu_1)^2}{2\sigma_1^2} \right] + (1 - p) \times \frac{1}{\sigma_2 \sqrt{2\pi}} \exp \left[\frac{-(x - \mu_2)^2}{2\sigma_2^2} \right] \quad (17)$$

Here, the p represents a weighting factor for the combination of single normal distributions. Additionally,

if the data is radially distributed, a linear combination Rayleigh distribution is shown in Equation (18).

$$F(R) = p \left(1 - \exp \left[\frac{-R^2}{2\sigma_1^2} \right] \right) + (1-p) \left(1 - \exp \left[\frac{-R^2}{2\sigma_2^2} \right] \right) \quad (18)$$

Consider the example previously presented in Figure 10. Now, we evaluate the same test data for fit to a double normal distribution as in Equation (17). The resulting PDF and CDF are shown in Figure 11.

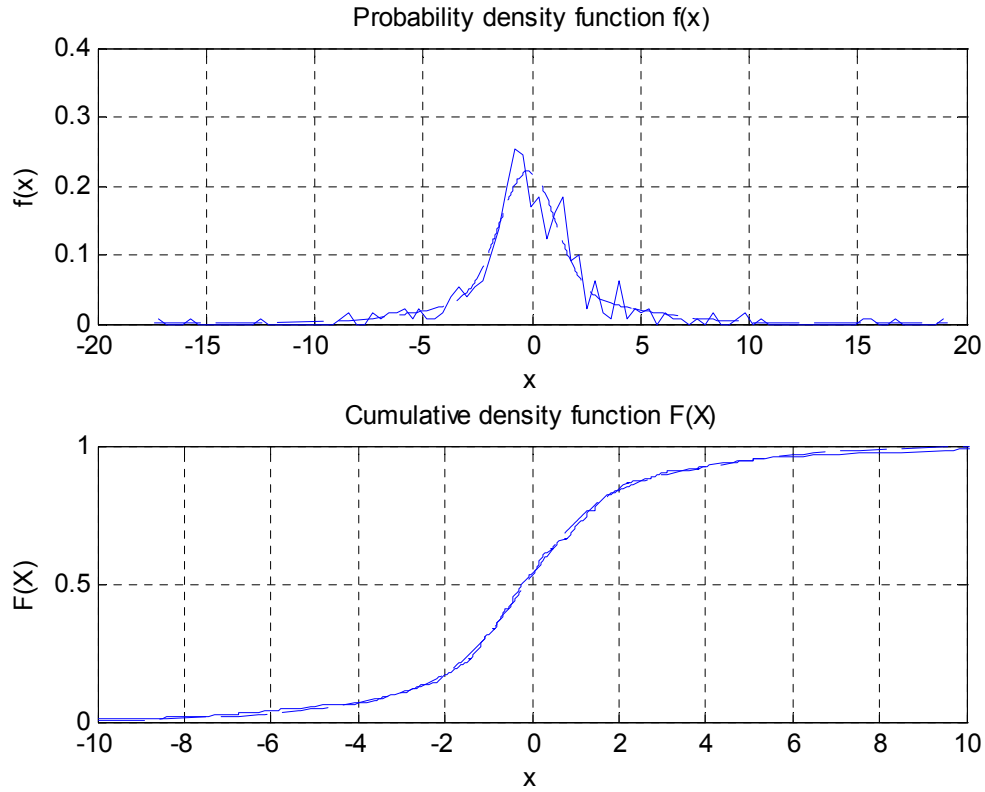


Figure 11. Double Normal fit to test data

We see here that the linear combination of two normal distributions fits the test data with accuracy and passes

the K-S test for confidence of fit with ninety-five percent confidence.

E. KOLMOGOROV-SMIRNOV (K-S) TEST

The K-S test is used to compare test data to a predicted distribution containing a set of parameters (μ_1 , μ_2 , σ_1 , σ_2 , p in the double normal CDF case) and determine the quality of fit. Given the statistical parameters used to generate an approximate distribution, we can create a set of data points derived from this CDF. We can then compare the actual test data to the data generated from our predicted CDF using MATLAB's *kstest* function. The *kstest* function returns a confidence interval in which the data is approximated by using hypothesis testing [5]. A limitation to the K-S test is that while it is very sensitive around the median value of the data, it is significantly less sensitive at the tails [2]. Based on this, many statisticians prefer to use the Anderson-Darling test of good fit; but it is only valid for a few specific distributions. Anderson-Darling cannot be applied to linear combinations of distributions, which negates its usefulness here [2].

F. LEGACY METHODS

Up until this point, the statistics and analysis provided assumed that the miss distances can be fit to a normal or radial distribution or a linear combination of these distributions. There is an additional method, called the probability of hit and probability of near miss ($P_{\text{HIT}}/P_{\text{NM}}$) method, that does not rely on this underlying

assumption [1]. Often, it is observed that a recorder might see the results shown in the Figure 12.

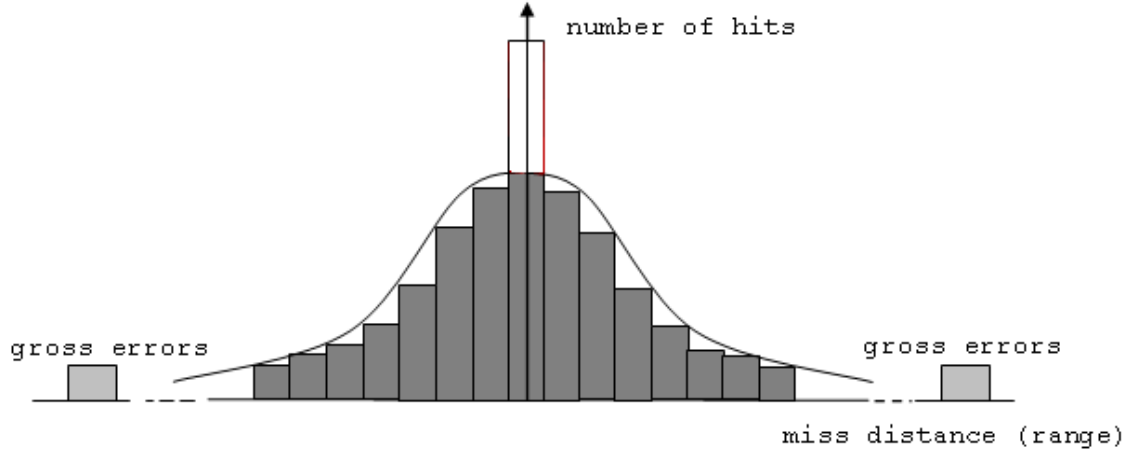


Figure 12. $P_{\text{HIT}}/P_{\text{NM}}$ Methodology (from [1])

This data is characterized by a large number of direct hits, a fair number of significant misses, and the remaining data appears to follow a normal distribution. The analysts of this data set calculated three values: probability of hit (P_{HIT}), probability of near miss (P_{NM}) and a CEP. The P_{HIT} is the number of data points in the unshaded region of Figure 12 divided by the total number of shots. The P_{NM} is the number of data points under the normal distribution after the gross errors are removed divided by the total number of shots. Gross errors are defined for this method as any data point that lies outside of the $\pm 4\sigma$, where the standard deviation used for eliminating errors is the common standard deviation given in Equation (11). This process is also iterative, as once the common standard deviation is calculated and the data points outside of 4σ are removed from the radial miss distance vector (and the range and deflection vectors corresponding to the same

radial data point), a new common standard deviation of the remaining data points must be calculated. The remaining data set is then compared to this new 4σ threshold to ensure no new gross errors exist. Thus, if we represent the probability of gross error by P_{GE} , we can numerically calculate the P_{GE} as the total number of errors removed divided by the total number shots. Finally, a CEP is calculated as it only relates to the data for the P_{NM} (normally distributed region). Therefore, the total probability can be found by Equation (19).

$$P_{NM} + P_{HIT} + P_{GE} = 1 \quad (19)$$

A more detailed description of this algorithm can be found in [1] and its exact implementation using MATLAB is included in the Appendix.

IV. TOOLBOX DEVELOPMENT

It is desired among the weaponeering community to have a standard set of tools to aid in evaluating delivery accuracy of weapon systems. Although the techniques that exist are accurate, we have shown in the previous example that a linear combination of distributions seemingly fits test data with high confidence [1]. Without the use of computing software, identifying the combination of distributions would be impossible. The tools depicted in Figure 13 were created to do just this.

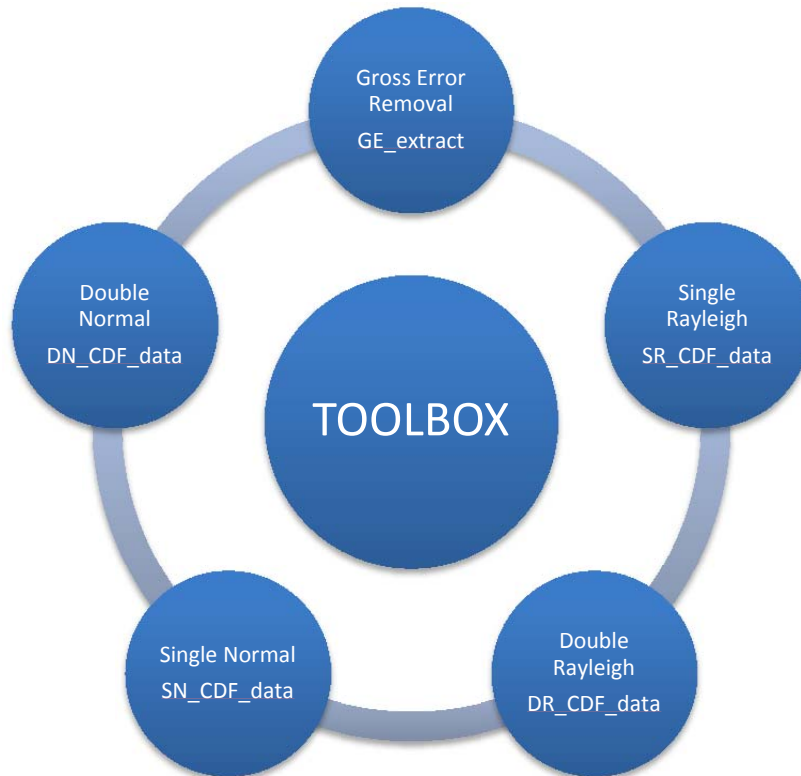


Figure 13. The Toolbox

A. GROSS ERROR EXTRACTION FOR RANGE AND DEFLECTION

Often in weapons applications it is desirable to evaluate data in range and deflection. In fact, if you are given data that is radial and you have the offset angle for each occurrence, it is appropriate to calculate range and deflection miss distances to add to the fidelity to the sample. Given a text file (which can easily be generated using cut and paste from Excel or any other tabular software) in which the range and deflection miss distance are in two column vectors, a user can run the MATLAB script `GE_extractXY` to remove the gross errors from the data sets. A gross error is defined as any data point outside of 4 standard deviations from the mean. The `GE_extractXY` program will identify a gross error in range and remove the corresponding data point in deflection and vice versa. When running `GE_extractXY`, the user will be prompted to identify the text file that contains the data. The program will then output three files to the same directory in which it is saved. The first file is called `GE_extractXY.txt` and contains the statistical results for the data provided. The program calculates mean and standard deviation for each vector and also displays the number of gross errors that were removed. The second file is titled `GE_extract_output_range.txt` and contains the resulting range vector with the gross errors removed. Similarly, the third file titled `GE_extract_output_deflection.txt` contains the resulting deflection vector with gross errors removed. Finally, a popup will contain a graph of impact points with the gross errors removed. Figure 14 illustrates the input and output to the `GE_extractXY` algorithm.



Figure 14. GE_extractXY Input/Output

B. DOUBLE NORMAL DISTRIBUTION

The next program in the toolbox is a MATLAB script called DN_CDF_data. This program takes in a text file with a single column vector and returns the statistics for fitting the data to a double normal distribution. When running the program, the user will be prompted to identify which text file contains the data desired for evaluation. The program will then perform two operations. First, a popup will show the PDF and CDF of the double normal curve fitted to the data with the raw data superimposed. Second, a file titled DN_data.txt will be saved in the same directory in which the program was run and contains the statistical results for the analysis. This program also contains a gross error removal algorithm. If you have run GE_extractXY.exe on the data set, the gross error statistics will all be zero. The other statistics in the DN_data.txt file are the mean and standard deviation of each normal distribution and the weighting factor relating the linear combinations of normal distributions. The confidence interval is calculated using the K-S test method. Figure 15 illustrates the input and output to the DN_CDF_data algorithm.



Figure 15. DN_CDF_data Input/Output

C. DOUBLE RAYLEIGH DISTRIBUTION

The next program in the toolbox is a MATLAB script called `DR_CDF_data`. This program takes in a text file with a single column vector and returns the statistics for fitting the data to a double Rayleigh distribution. Clearly, this assumes that your vector is radially distributed and you are unable to calculate range and deflection miss distance due to a lack of information. When running the program, the operator will be prompted to identify the text file containing the data for evaluation. Once complete, the program will display a visual graph of the PDF and CDF of the curve fitted to the data with the raw data superimposed. Second, a text file called `DR_data.txt` will be saved in the same directory in which the program was run and contains the statistical results from the analysis. This program also contains an error removal algorithm to remove gross errors, as it is not possible to run `GE_extractXY` in this case. The first section in `DR_data.txt` will contain statistics on the number of gross errors removed and the second section will contain the mean and standard deviation of each Rayleigh distribution and the weighting factor relating those distributions. To reiterate a concept from Chapter II, the error removal algorithm is based on the common standard deviation in range and deflection (σ_c), not on the standard deviation of the radial miss distance (σ_r). In this case, we

only pass the radial miss distances into the algorithm and exploit the relation that $\sigma_r=0.655\sigma_c$ to remove the errors. The confidence interval is calculated using the K-S test method. Figure 16 illustrates the input and output to the DR_CDF_data algorithm.



Figure 16. DR_CDF_data Input/Output

D. SINGLE NORMAL DISTRIBUTION

This program will evaluate a column vector for fit to a single normal distribution. Running the MATLAB script titled SN_CDF_data will prompt the user to input a text file containing a single column vector. Once complete, the program will pop up a visual graph of the PDF and CDF of the curve fitted to the data and with the raw data superimposed. The program will also save a text file, called SN_data.txt, to the same directory in which the program was run. This file will contain the statistics for the gross errors that were removed from the sample as well as the statistics for the remaining distribution. Specifically, these statistics are the mean, standard deviation and confidence interval calculated by the K-S test function in MATABL. Figure 17 illustrates the input and output to the SN_CDF_data algorithm.



Figure 17. SN_CDF_data Input/Output

E. SINGLE RAYLEIGH DISTRIBUTION

Much like the single normal program above, this program will evaluate an inputted vector for fit to a single Rayleigh distribution. The MATLAB script titled `SR_CDF_data` will prompt the user to input a text file containing a single column vector. Once complete, the program will pop up a visual representation of the PDF and CDF of the curve fitted to the data with the raw data superimposed. The program will also save a text file to the same directory in which the program was run called `SR_data.txt`. This file will contain the statistics for the gross errors that were removed from the sample based on the common standard deviation as well as the statistics for the remaining single Rayleigh distribution. These statistics include the mean, standard deviation and a confidence interval based on the K-S test function in MATLAB. Figure 18 illustrates the input and output to the `SR_CDF_data` algorithm.



Figure 18. `SR_CDF_data` Input/Output

F. MATLAB'S COMPILER DEPLOYMENT TOOL

As licenses to MATLAB are not inexpensive, we will exploit some of the functionality in the MATLAB software to provide the tools described above to all users. The compiler tool (called `deploytool`) converts MATLAB scripts to C code. It then packages the C code in the form of an executable file that can be run on any machine. The tool

also uses free software titled MCR_installer that must be installed before running the executable file and allows some specific functions of MATLAB to operate [5]. The free software can be found for all operating systems on the Math Works website. As the MATLAB users will find little usefulness in the text files that are output from the code segments because they can see the data directly in MATLAB, the text files that are generated by the code are required for the users of the executable files.

It is important to note that installation of the MCR_installer file will require administrator privileges to your computer and it will change the registry file.

THIS PAGE INTENTIONALLY LEFT BLANK

V. TOOLBOX IMPLEMENTATION FOR ACCURACY ASSESSMENT

This section describes how to use the code segments described in the previous chapter to analyze data as it applies to calculating delivery accuracy. The code is written to always give the analyst a solution so the results should be closely scrutinized.

A. DATA IDENTIFICATION

The first step is to identify the type of data that you are evaluating. If you are given radial miss distances with an offset angle, the data should be converted to range and deflection miss distances for analysis. If you have range and deflection data, continue with it in this form. The only other possibility is to have radial miss distances. The reason range and deflection miss distance is preferred is to enhance the fidelity of the statistics used to describe the data. Mainly, radial data assumes that the standard deviations in range and deflection are equal, which is not always a good assumption. As mentioned previously, common to artillery firings, there is a relatively small dispersion in deflection and a much larger dispersion in range [3].

B. RANGE AND DEFLECTION DATA

1. GE_extractXY

The first tool to run both range and deflection vectors through is GE_extractXY. This code will remove the gross errors from each vector and the corresponding data point in the opposite vector. To illustrate the importance of this step, consider a weapon that lands on the range

axis but misses the positive deflection axis by more than 4 standard deviations. If you evaluate the range data independently, the event would appear as a direct hit, whereas it would be considered a gross error in deflection. This data point is clearly a miss and should be removed based on our definition of a gross error. GE_extractXY will output two text files, one containing the range vector and one containing the deflection vector with gross errors removed. The resulting vectors are inputted into SN_CDF_data separately.

2. SN_CDF_data

Given a text file with a single vector, SN_CDF_data will return the statistics σ and μ , and the confidence of fit to a single normal distribution. If the fit does pass the confidence test, the resulting σ can be used to calculate REP or DEP, respectively, using Equations (13) and (14). Clearly, if the data does not pass the confidence test within an acceptable limit (normally ninety-five percent), the statistics calculated will hold no value and the fit to a double normal distribution should be evaluated. This process should be implemented twice; once for range and once for the deflection vectors.

SN_CDF_data does contain a gross error removal algorithm. It will only look for errors in the inputted single vector. This was included to increase the functionality of the code. When analyzing range and deflection as a process, GE_extractXY should always be utilized first.

3. DN_CDF_data

Given a text file with a single vector, DN_CDF_data will return the statistics σ_1 , σ_2 , μ_1 , μ_2 , p and the confidence of fit to a double normal distribution. Clearly, if the data does not pass the confidence test within an acceptable limit (normally ninety-five percent), the statistics calculated will hold no value. If the fit does pass the confidence test, the analyst should pay careful attention to the results for p , σ_1 and σ_2 . In some cases, the weighting factor (p) that is calculated can be less than ten percent or even negative. This shows that your data is more likely to be from a single normal distribution and not a double normal distribution. In addition, for small values of p , the standard deviations can often vary by 2 orders of magnitude. This is another sign that your data is more likely to fit a single normal distribution. However, if the weighting factor p holds significant weight and the data passes the confidence of fit test, the values of σ_1 and σ_2 can be used to calculate two error probable values. Additionally, you can identify a CDF that models the data from which the input was generated. This process should be implemented twice, once for the range vector and once for the deflection vector.

C. RADIAL DATA

1. SR_CDF_data

If you are not given data in range and deflection, but rather radial miss distances, the single vector should be inputted into SR_CDF_data. In this case, the gross error removal algorithm is implemented in the code and the results will be output with the σ and result of confidence of fit test in the SR_data text file. If the data passes the confidence test, the σ is used to calculate a CEP using Equation (15). If the data fails the confidence test, we analyze the data set for fit to a double Rayleigh distribution.

2. DR_CDF_data

Given radial data, DR_CDF_data removes the gross errors from the data set and returns σ_1 , σ_2 , and p . The code also returns the confidence of fit to a double Rayleigh distribution in the DR_data text file. If the data passes the confidence test, σ_1 and σ_2 are used to calculate two independent CEP values and p is used to weight them accordingly. The exact implementation of this method will be described in the next chapter.

As mentioned previously, the analyst should pay close attention to the statistical outputs to ensure they are logical. The code will always produce a solution, but the values returned may not make mathematical sense.

A flow chart for the single distribution process can be found in Figure (19).

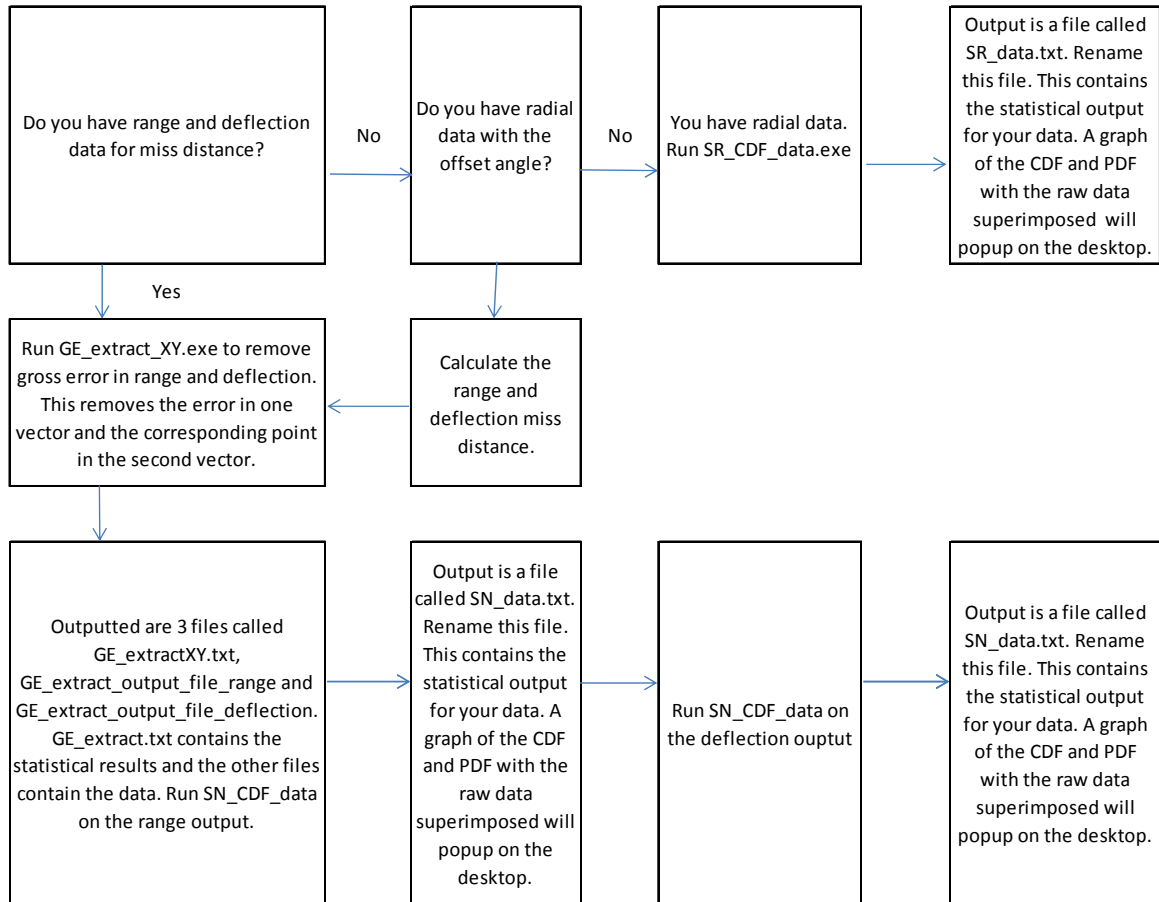


Figure 19. Single Distribution flow chart

The same flow chart can be used when evaluating for fit to a linear combination of distributions. For the range and deflection case, you will run the DN_CDF_data program and the output will be DN_data.txt. Similarly, in the radial case, you will run DR_CDF_data and the output will be DR_data.txt. It is recommended that you rename the text file that contains the statistical output as the program will re-write to the DR_data.txt file when you run the program a second time.

D. ALTERNATIVE APPROACH

In an effort to produce a more accurate solution, a less intuitive approach can be used to evaluate the data. In this case, fit to a linear combination of distributions is evaluated first as it may more precisely describe impacts. In practice, this would reverse the process described above and you would evaluate DN_CDF_data first and then SN_CDF_data. Similarly, if given radial data, you would analyze DR_CDF_data and then SR_CDF_data. Computing power has resulted in the more frequent use of mixture models and although not historically implemented, the precision of the solution will be better.

VI. CALCULATING WEAPON EFFECTIVENESS

Although the tools described in the previous chapter provide valuable statistics concerning the accuracy of a weapon system, this is not the end goal. The error probable values are used to not only describe the accuracy, but more importantly the accuracy is used to calculate the effectiveness of a weapon or the probability of kill. Given a large bomb, this solution may seem trivial in that a near miss of a few feet from the intended target is insignificant. However, highly precise munitions that carry very little explosive material are intended to not wipe out a city, but rather kill a very specific target. Often, a kill in this case not only depends on *if* we hit the target, but *where* we hit the target. The precision not only affects the probability of kill, but also collateral damage estimates.

A. PROBABILITY OF KILL CALCULATIONS

There are two methods that we will describe as a basic way to calculate the probability of kill. The first method uses a Monte Carlo simulation and the second exploits the mathematical convenience of the expected value theorem to calculate the probability of kill.

B. MONTE CARLO APPROACH

A Monte Carlo simulation runs an iteration of a desired calculation a series of times until the final result converges. In this simple case, we will draw a random impact point from the CDF generated using the toolbox presented. For a fixed target size, we then check

to see if the impact point hits or misses the target. If we hit the target, we increase a counter; if we miss, we do nothing. At the end of the loop we will average the number of hits by the total number of random draws. Running this for a large number of iterations, we can determine the probability of hitting the target or probability of kill.

C. EXPECTED VALUE THEOREM

If we have a continuous random variable, x , the expected value of x is defined in Equation (20).

$$E(x) = \int_{-\infty}^{+\infty} x f(x) dx \quad (20)$$

Further, suppose that we have a function of a random variable $y = h(x)$ where x is derived from the PDF $f(x)$. In this case, we can define the expected value of y in Equation (21).

$$E(y) = E(h(x)) = \int_{-\infty}^{+\infty} h(x) f(x) dx \quad (21)$$

To calculate the probability of kill in this case, we perform the integration of Equation (21). For a normal distribution there is not an explicit solution to the integral so we will approximate the value of the CDF using the error function estimate. For the Rayleigh distribution there is an explicit result.

Using this method, we calculate the probability of kill for range and deflection separately. Because these samples are independent, we can exploit the property of independent probability calculations as in Equation (12) and get the total probability of kill by multiplying the

probability of kills from range and deflection together. In the radial case, the probability of kill returned is the total probability of kill.

D. P_{HIT}/P_{NM} METHODOLOGY

To calculate the probability of kill using the P_{HIT}/P_{NM} method, we exploit the fact that the data from which the hits and misses were calculated is independent. Mainly, the data in the sample is not double counted by either statistic. Thus, we calculate the probability of kill in Equation (22).

$$Pk_T = P_{NM} * Pk_1 + P_{HIT} * Pk_2 \quad (22)$$

Pk_1 is calculated using the CEP that corresponds with the normal distribution of P_{NM} . Pk_2 is calculated using a CEP of zero (a direct hit). Here P_{HIT} and P_{NM} can be seen as weighting factors.

E. THE JMEM WEAPONNEERING SYSTEM

Although we simulate the probability of kills using a hit or miss methodology, the Joint Munitions Effectiveness Manual Weaponneering System (JWS) contains complex data of actual weapons' blast radii, fragmentation data, etc., as well as the dependence on trajectory that the weapon attacks the target and is used to provide the most detailed probability of kill results. The JWS is limited in its input, as it currently requires a single REP, DEP, or CEP to generate effectiveness models. Because the double normal distribution produces statistics that are not independent, we cannot utilize this information as input to the JWS.

Some examples why the output from a double normal distribution cannot currently be used are described below.

1. Given σ_1 , σ_2 , μ_1 , μ_2 , and p one might try and weight the standard deviations to produce single standard deviation (σ_{NEW}). This is represented in Equation (23).

$$\sigma_{\text{NEW}} = p*\sigma_1 + (1-p)*\sigma_2 \quad (23)$$

Properties of normal distributions dictate that this result is a new single normal distribution and this distribution will not represent the data from which the original statistics were derived.

2. Given the σ_1 , σ_2 , μ_1 , μ_2 , and p for both range and deflection, one might try and input σ_1 corresponding to range into JWS with the σ_1 corresponding to deflection in the form of REP and DEP, respectively. Then, in a separate calculation, provide the JWS σ_2 from range data and σ_2 from deflection data in the form of a separate REP and DEP. Finally, you could use the weighting factor to weight the effectiveness much like the $P_{\text{HIT}}/P_{\text{NM}}$ methodology. The miss distances in range are independent and the deflection miss distance is also independent when evaluated separately. However, when combining the range and deflections miss distance for JWS calculations, independence cannot be assured. For example, there is no way to verify that the same data points used to calculate σ_1 in range are the same finite set used to calculate σ_1 or σ_2 in deflection.

What this dictates is that for delivery accuracy calculations, double normal distributions can be utilized and applied to range and deflection miss distances for accuracy analysis. When providing data to JWS however, the

data must be converted to radial miss distances and evaluated for fit to a double Rayleigh distribution. The double Rayleigh distribution will produce σ_1 , σ_2 , μ_1 , μ_2 , and p . By converting the data to radial miss distance, we remove the need to have independence between range and deflection axis and can provide JWS with two independent CEP values. We can then weight the resulting probability of kill (P_k) produced by JWS as in Equation (24).

$$P_{k_T} = P_{k_1} * p + P_{k_2} * (1 - p) \quad (24)$$

This method is very similar to that of the $P_{\text{HIT}}/P_{\text{NM}}$ methodology except that the P_{k_2} is calculated with an actual CEP value vice the CEP equaling zero in the $P_{\text{HIT}}/P_{\text{NM}}$ method.

JWS is being modified to accept the statistical output in the form of a double normal distribution. This change will be available to the user soon. The necessity for this change will be illustrated in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. IMPLEMENTATION FOR EFFECTIVENESS

The following process should be used to calculate error probable values to be used as input for effectiveness calculations. We will implement the recommended methodology that uses the linear combination of distributions to evaluate weapon accuracy as a primary method. Mainly, the double distributions will be examined first, followed by the single distributions for more complete analysis. This is done to illustrate the flexibility inherent to the toolbox and is made possible by the toolbox code.

A. RANGE AND DEFLECTION DATA

Given range and deflection miss distance, the analyst should start by removing gross errors using `GE_extractXY` from the range and deflection vectors. The resulting output with gross errors in range and deflection removed should be inputted into `DN_CDF_data` separately to see if they pass a confidence on fit to a double normal distribution. If both range and deflection data pass the fit test, the statistics will be able to be input into JWS in the future. As this is not yet supported, the resulting range and deflection vectors are input into `SN_CDF_data` first to see if they pass the confidence of fit to a single normal distribution. If both data sets pass, calculate both REP and DEP values, respectively, for input into JWS. If either distribution fails to fit a single normal distribution, the analyst should convert the data to radial miss distances.

B. RADIAL DATA

Whether the data failed the single normal (or double normal) distribution confidence of fit test or the analyst was only provided radial data, begin by imputing the vector into the DR_CDF_data algorithm. If the data passes the confidence of fit test for the double Rayleigh distribution, σ_1 and σ_2 are used to calculate CEP₁ and CEP₂. These values can be inputted into the JWS program separately. Thus, CEP₁ and CEP₂ will each have a probability of kill associated with them (Pk₁ and Pk₂). To find the total probability of kill, use the weighting factor to combine the probability of kills as described in Equation (24).

In the event that the data fails to pass the confidence of fit to a double Rayleigh distribution, the radial data should be input into the SR_CDF_data to determine if it fits a single Rayleigh distribution. If it passes the confidence of fit test, a single standard deviation output by the algorithm and is used to calculate a CEP value. This value is passed into JWS to determine a probability of kill. A flow chart of the entire process is shown in Figure 20. This will require modification once JWS supports a double normal distribution.

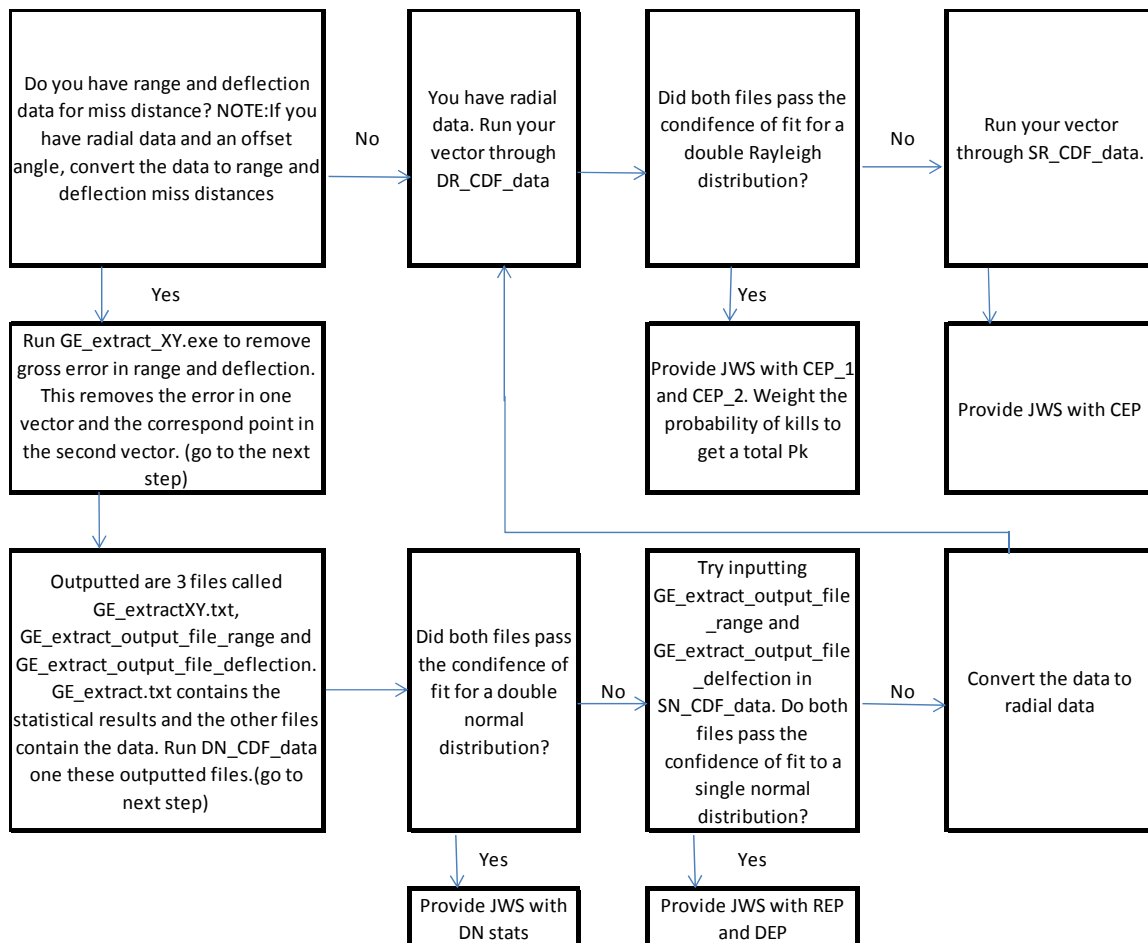


Figure 20. Effectiveness flowchart

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. COMPARISON OF DATA ANALYSIS METHODS AND THEIR EFFECT ON PK

To illustrate the need for the toolbox and its functionality, we investigate various accuracy methodologies applied to a single data set. By comparing the probability of kill values for each method, it will be clear why the toolbox was created. Mainly, the use of linear combinations of distributions significantly increases the precision in modeling the accuracy of test data, which directly impacts the probability of kill calculation.

A. PROBABILITY OF KILL

A Monte Carlo method may be used to calculate the probability of kill. To implement this we create a fixed target size. In this example, we used a square target that is centered about the origin. Then to calculate the probability of kill given a CEP value, we draw a random number from a single Rayleigh distribution characterized by the standard deviation associated with the CEP. Assuming that the origin is our aim point (the center of the target), the random number drawn is the radial miss distance for a single shot. We check to see if the radial miss distance is within the predetermined target size, and if so, we call it a hit. If the radial miss distance is outside the target size, the occurrence is a miss. Then, we average the number of hits by the total number of shots and this determines the probability of kill.

Similarly, if we are given REP and DEP, we draw a random number from a single normal distribution

characterized by REP and a second random number from a single normal distribution characterized by DEP. We then check to see if the miss distance in range and deflection falls within the target size. If the actual impact point is within the target dimensions, we count it as a hit and if the impact is outside the target size, it is a miss. Again, averaging the number of hits over the total number of shots results in the probability of kill.

To utilize a Monte Carlo method for a double normal distribution requires only a slight modification to the single normal method described above. In this case, the range misses are characterized by 2 independent single normal distributions given by REP_1 and REP_2 , and as such we randomly sample from each distribution according to the weight factor p . This is accomplished by creating a range vector constructed of 10,000 miss distances. The weighting factor p is used to populate this vector according to REP_1 and DEP_1 . Thus, $10000 * p$ elements are from a distribution generated from REP_1 and $10000 * (1 - p)$ elements are from the distribution corresponding to REP_2 . This creates a sample size of 10,000 miss distances generated randomly from the same distributions described by the double normal result. In deflection, there is a second set of 2 unique single normal distributions given by DEP_1 and DEP_2 that describe the deflection miss distances. We randomly sample the deflection miss distance according to p for this linear combination as well. This is done using the same method for the described for the range vector. Finally, given the 2 vectors of 10,000 elements each, we check to see if the randomly generated miss distance is within the target size

and count the number of hits. The probability of kill is the total number of hits divided by the number of shots.

The methods described above are then repeatedly applied as we increase the size of the target. The results in a range of probability of kill values over various target sizes, defined to be the length and width of the target.

B. MEDIAN RADIAL ANALYSIS

Given a tabular data set of 364 miss distances shown graphically in Figure 21, we want to calculate a CEP value. This represents the simplest and faster way to generate an accuracy statistic.

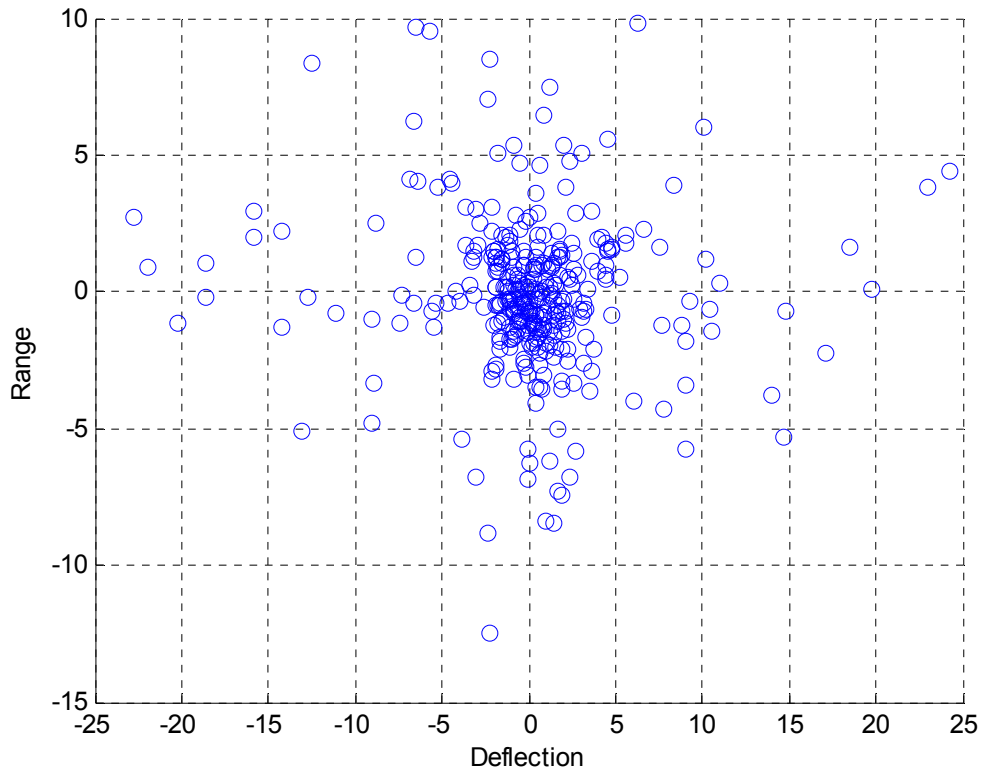


Figure 21. Test data impact points

The process to calculate a CEP is listed below.

1. Take the range and deflection miss distances and calculate radial miss distances in Excel.
2. Rank order the misses and calculate the median value.
3. This value directly returns the CEP.
4. In this case, CEP = 2.465.

We then implement the Monte Carlo method to calculate the probability of kill over various square target sizes resulting in Figure 22.

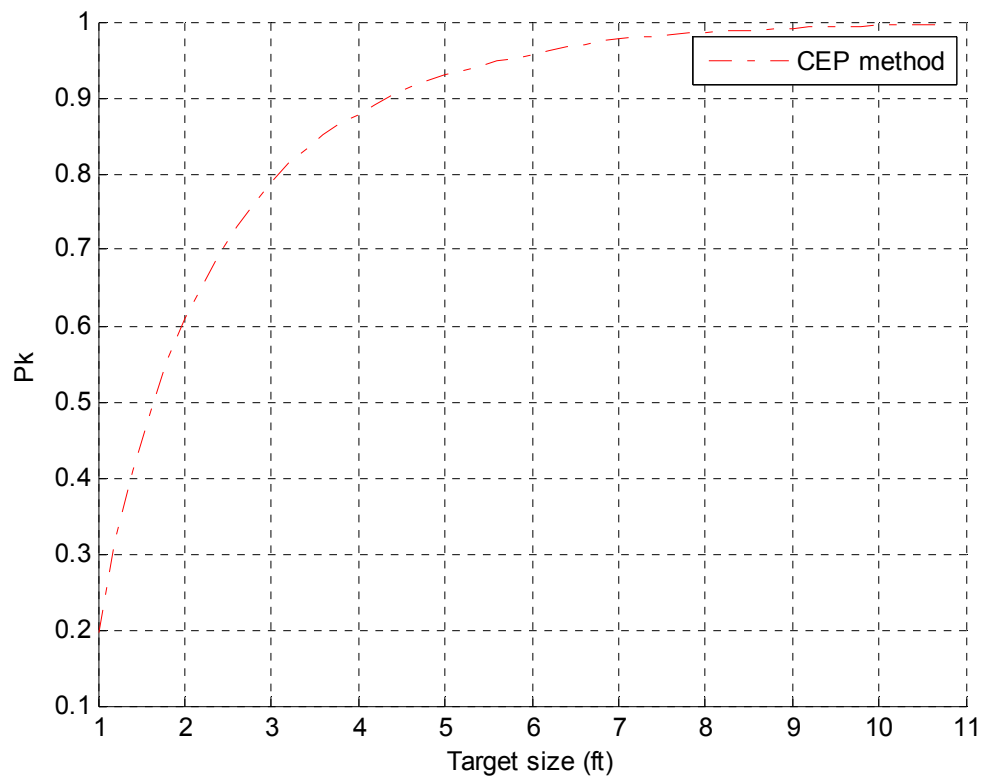


Figure 22. P_k based on CEP

C. ANALYSIS IN RANGE AND DEFLECTION

In an effort to improve the analysis process using the same data set, we now keep the data in range and deflection separate. We remove 19 gross errors that lie outside $\pm 4\sigma$ from the data set. This is described in Chapter III, Section A. We then assume that the remaining data is from a single normal distribution and calculate the standard deviation in range ($\sigma_R=4.11$) and the standard deviation in deflection ($\sigma_D=2.59$). These correspond to a value of REP = 2.77 and DEP = 1.75, respectively. A Monte Carlo method was implemented by sampling REP and DEP and the probability of kill was calculated again for different target sizes. Figure 23 plots the CEP method compared with the REP and DEP method. As expected, maintaining the data in REP and DEP increases the precision of the model. In this case, the standard deviation in range is almost double the standard deviation in deflection. Thus, assuming the standard deviations are equal as in the radial case is just not as accurate. The disparity in Pk values is troublesome as they differ by almost 40 percent in some cases. This is what motivates the toolboxes design.

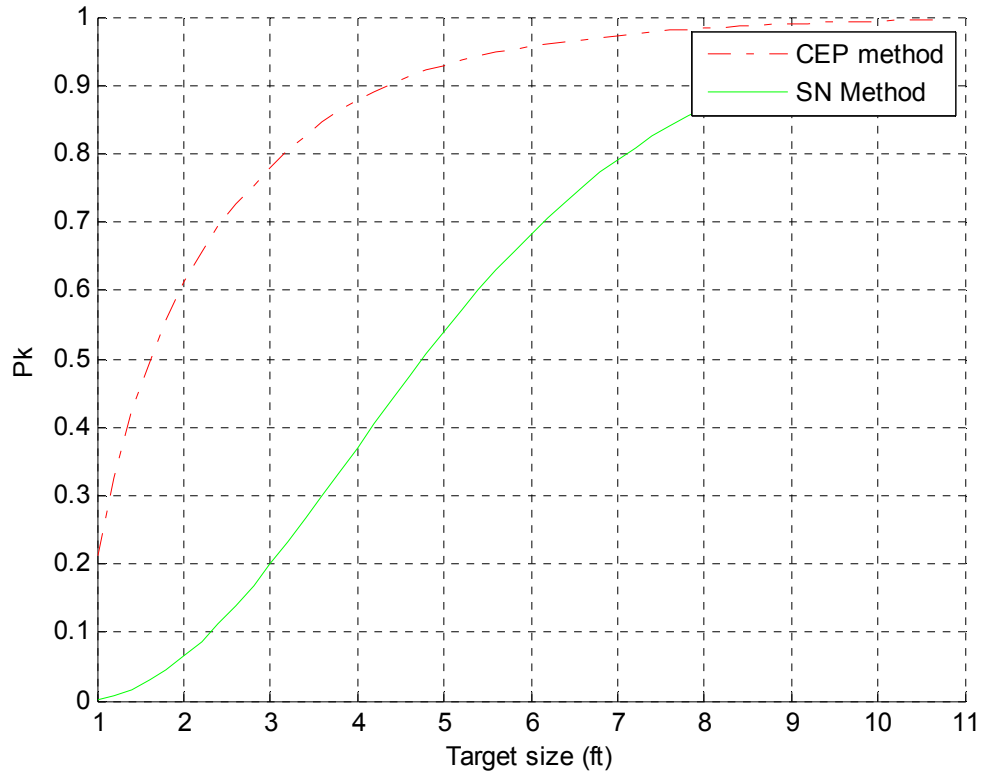


Figure 23. Comparison: CEP vs. REP/DEP

D. THE TOOLBOX-DOUBLE NORMAL ANALYSIS

Considering the same data set, we remove the blind assumption that the data fits a single normal distribution. In fact, using the SN_CDF_data program within the toolbox, the data set that remains after the 19 gross errors are removed fails to pass a confidence of fit at even a 60 percent confidence level. Thus, we pass the impact points in range and deflection into the DN_CDF_data program separately. The statistical results are shown in Table 1 and the plots of the PDF and CDF with data superimposed are shown in Figures 24 and 25.

DN_CDF_data								
Range					Deflection			
p	0.694				p	0.615		
$\mu 1$	0.137	$\sigma 1$	1.653		$\mu 1$	-0.242	$\sigma 1$	1.346
$\mu 2$	0.45	$\sigma 2$	7.91		$\mu 2$	2.680	$\sigma 2$	3.953

Table 1. DN_CDF_data results

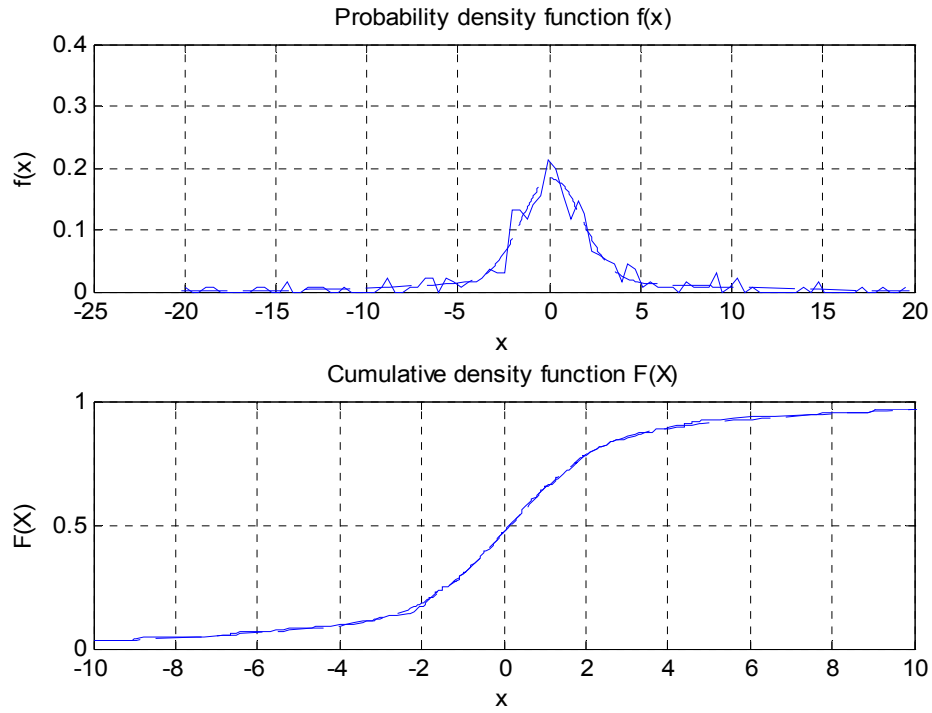


Figure 24. DN_CDF_data graph for range

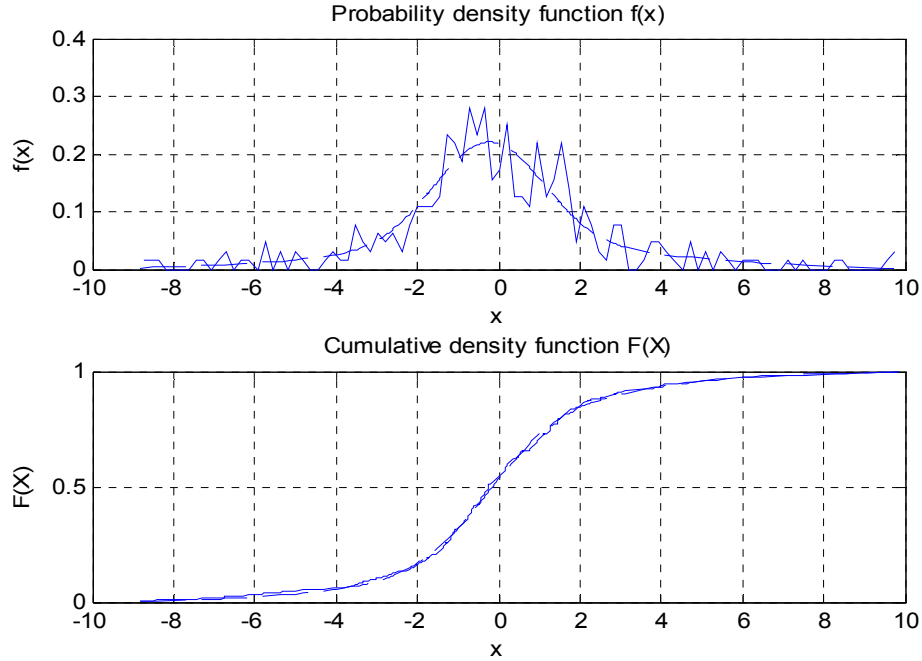


Figure 25. DN_CDF_data graph for deflection

The data passes the confidence of fit at a 95 percent confidence level to a double normal distribution in range and deflection. Given the results, we calculate the probability of kill by randomly sampling a linear combination of distributions according to the weighting factor in range. This method was described in earlier in this chapter and involved creating a vector of 10,000 elements from which a proportion p are generated from REP_1 and $1-p$ are generated from REP_2 . Similarly, we execute the same random sampling in deflection based on DEP_1 and DEP_2 . Given this randomly generated impact point, we check to see if the shot hits the target. Averaging the number of hits by the total number of shots at the target, we have the probability of kill for a fixed target size. We then increase the target dimensions to generate probability of kill values for various target sizes. Finally, to compare

the double normal method to the other methods presented, the probability of kill is plotted in Figure 26 for the double normal method, single normal method, and the radial method.

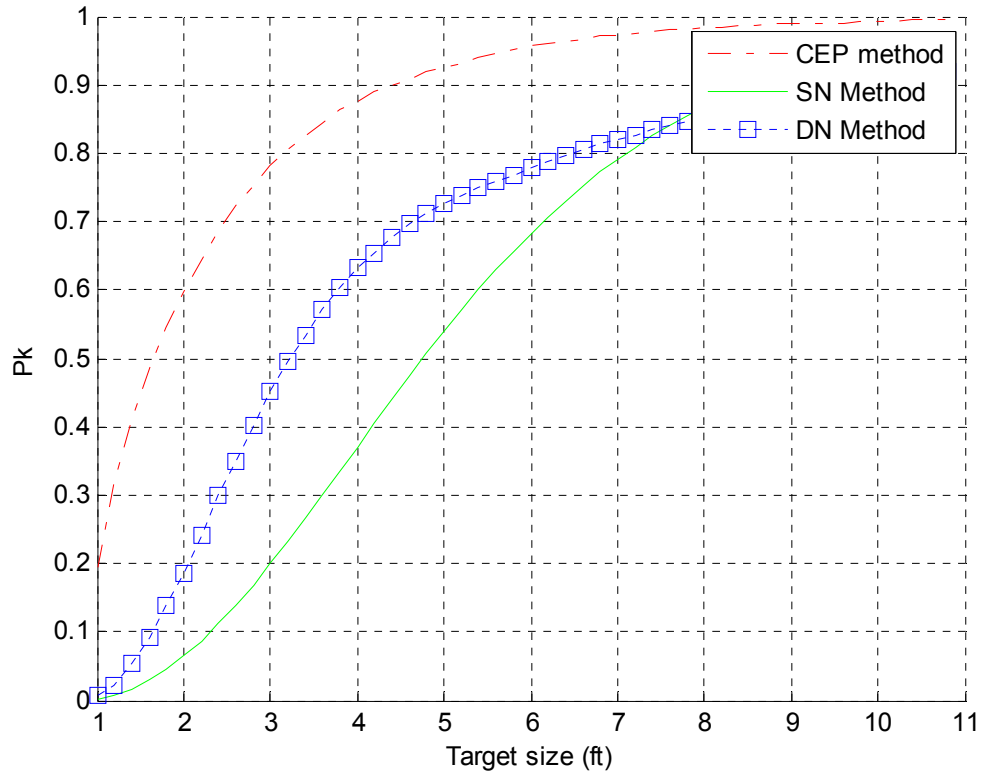


Figure 26. Comparison: DN method/SN method/Radial method

In comparing the 3 methods, we see that the radial method significantly over-estimates the Pk value for all target size. The assumption that the standard deviation in range and deflection are equal significantly reduces the accuracy of this method. The single normal method underestimates the Pk value for most target sizes as the data set is not modeled well by a single distribution. The double normal method made possible by the toolbox allows

for the most accurate modeling of the data set. This directly affects the precision of the probability of kill calculations. This impact can be realized when considering that mission planners will base a mission on a specific probability of kill for a target. They may launch multiple weapons at a target to achieve a 90 percent Pk value. The differences in the methods describe can result in too many or too few weapons being launched. This could directly result in a mission failure or in extra, unneeded weapons being launched costing taxpayer money and reduced inventories for future missions. In the case of air launched weapons, this directly puts more pilots in harm's way.

IX. FURTHER EXAMPLE TOOLBOX APPLICATIONS

In this section, we present more examples to illustrate the need for not only a standardize method of calculating weapon accuracy, but we will also further illustrate the need for implementing linear combination techniques presented in the previous chapters.

A. CASE STUDY 1

1. Calculating Delivery Accuracy

Using the techniques previously presented, consider the miss distance data given in Table 2. The data was originally presented as range and deflection miss distances and the radial miss distances were calculated for later use in Excel. First we will run the GE_extractXY program to remove gross errors in range and deflection. Remember, that if the code identifies an error in range, it will remove the corresponding data point in deflection and vice versa. Out of the 76 data points, 2 gross errors are removed. Figure 27 shows the plot of the remaining data set with gross errors removed.

Weapon A						
Range miss distance	Deflection miss distance	radial miss distance		Range miss distance	Deflection miss distance	radial miss distance
-10.0306	1.6526948	10.17		-6.48854	1.746882	6.72
-15.6853	-0.4882768	15.69		-3.4803	-6.611925	7.47
-9.65046	-10.123535	13.99		4.053778	-5.125377	6.53
-12.0806	2.6809026	12.37		29.44539	-8.866431	30.75
49.56104	-8.0855295	50.22		14.28569	0.1115436	14.29
4.114559	8.8852719	9.79		12.03432	-15.81656	19.87
-1.86632	3.8594538	4.29		3.378498	-4.015787	5.25
4.895866	9.2970042	10.51		5.099985	0.98552	5.19
9.47692	-12.644661	15.8		-1.75845	-0.73549	1.91
3.305097	-8.5139389	9.13		4.382794	-2.547982	5.07
17.71039	-8.4384881	19.62		1.730302	3.9348632	4.3
10.26875	3.8510995	10.97		9.059405	-6.029285	10.88
1.056869	11.412715	11.46		-1.75738	-0.738047	1.91
5.201057	8.5517761	10.01		62.85378	-16.49392	64.98
2.398486	4.5658221	5.16		37.26428	-3.470591	37.43
-17.373	-2.3216259	17.53		33.67541	-0.724638	33.68
12.62246	15.577345	20.05		3.840316	2.1258474	4.39
-7.12678	-4.7807452	8.58		12.97877	11.143522	17.11
-7.08519	5.2066772	8.79		11.79049	8.7898803	14.71
-7.19539	-19.717046	20.99		28.6502	9.6613791	30.24
-2.26301	-5.9249476	6.34		20.51767	15.242599	25.56
-2.83356	-9.6677223	10.07		-12.0626	-15.9177	19.97
13.65589	-9.2653143	16.5		-13.004	2.6486669	13.27
-9.2706	6.7848205	11.49		9.000507	9.1755477	12.85
0.478825	-1.8145258	1.88		-27.5337	-24.96625	37.17
7.579317	-9.5516601	12.19		-2.55039	-20.86326	21.02
63.25171	-30.180401	70.08		-24.6884	-27.18075	36.72
-5.07739	3.3123543	6.06		-30.1314	-5.296147	30.59
1.036115	-2.4797135	2.69		-10.3493	4.5357514	11.3
20.63737	-15.364597	25.73		-23.1435	-12.2365	26.18
3.439338	-5.5705593	6.55		51.59975	-42.89222	67.1
6.280177	2.3233214	6.7		101.9149	-69.09114	123.13
0.080233	-3.6765674	3.68		116.3996	-67.61015	134.61
-4.46198	4.7067751	6.49		74.2588	-29.5293	79.91
6.516216	3.9661641	7.63		15.2239	-0.482488	15.23
-11.4414	0.1875602	11.44		19.07191	-7.567178	20.52
-5.06205	-7.9414955	9.42		15.02195	-2.709418	15.26
-6.97965	2.7558156	7.5		3.118798	-5.222366	6.08

Table 2. Weapon A data

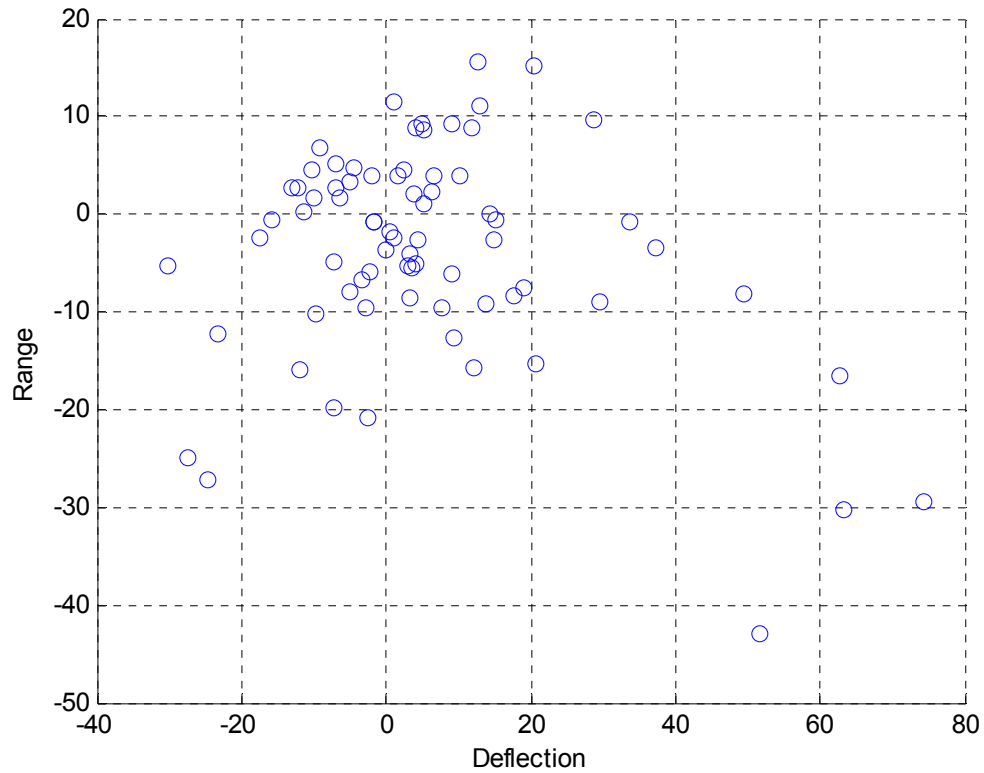


Figure 27. Range and deflection plot for Weapon A

The remaining 74 data points in range are returned in `GE_extract_output_file_range` text file and similarly the seventy-four data point in deflection are outputted to `GE_extract_output_file_deflection` text file. We now run the output files through `DN_CDF_data` individually. The statistical results are shown in Table 3.

DN_CDF_data								
Range					Deflection			
p	0.404				p	0.933		
μ_1	7.666	σ_1	28.961		μ_1	-2.635	σ_1	8.286
μ_2	1.491	σ_2	8.424		μ_2	1.18E+09	σ_2	1.98706E+12

Table 3. Weapon A - DN_CDF_data results

As seen from in Table 3, the range miss distance is modeled extremely well by a double normal distribution, but the deflection miss distance is not. Figures 28 and 29 show the plot of the PDF and CDF as well as the test data superimposed.

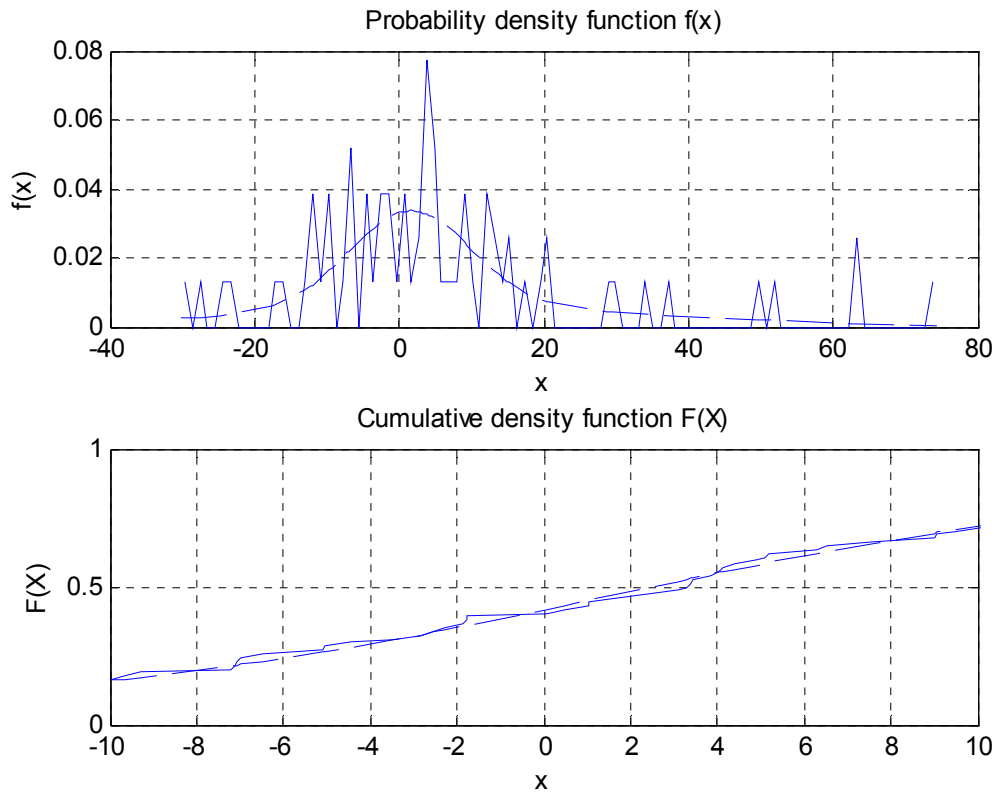


Figure 28. Weapon A - DN_CDF_data graph for range

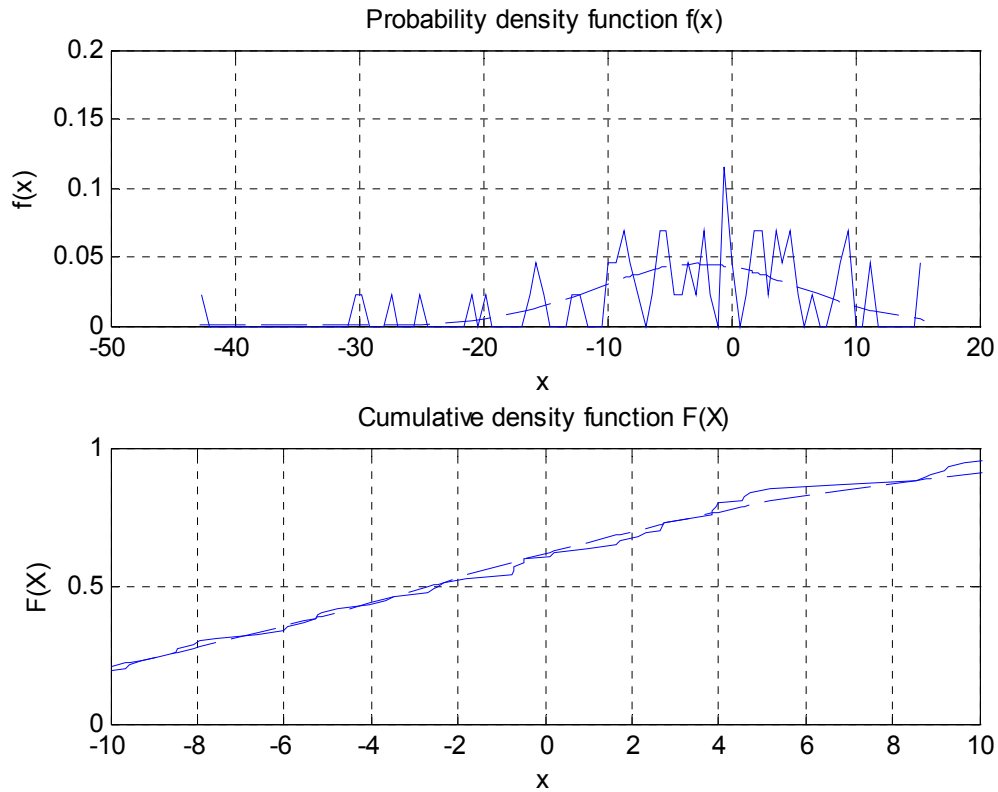


Figure 29. Weapon A - DN_CDF_data plot for deflection

Although the deflection data passes the 95 percent confidence of fit test, the weighting factor is not strong and the mean and standard deviations are magnitudes apart when comparing the first normal distribution in deflection to the second distribution of the linear combination. This is an example of the toolbox giving the analyst a solution, but one that is not valid. The analyst must scrutinize the results to ensure they are reasonable. Thus, we input the range and deflection miss distances to the SN_CDF_data program. The range data passes the fit to a single normal distribution with $\mu = 5.868$ and $\sigma = 19.804$. Similarly, the deflection data passes the confidence of fit with $\mu = -3.688$ and $\sigma = 10.915$ to a single normal distribution.

The plots of the PDF and CDF are shown in Figures 30 and 31, respectively.

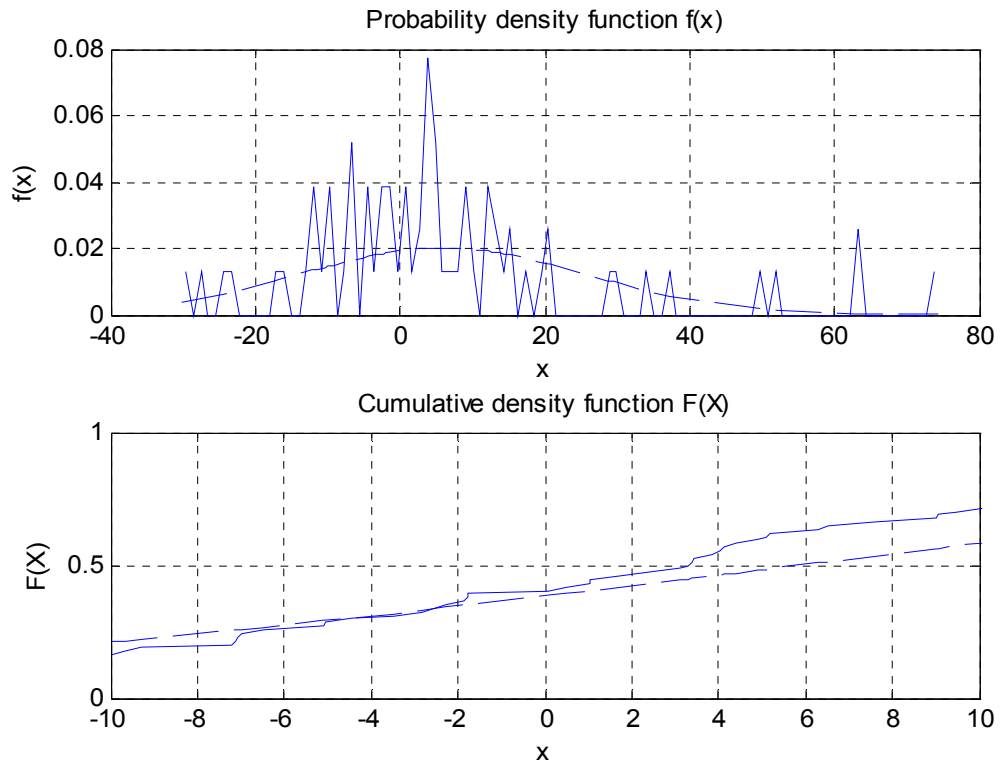


Figure 30. Weapon A - SN_CDF_data for range

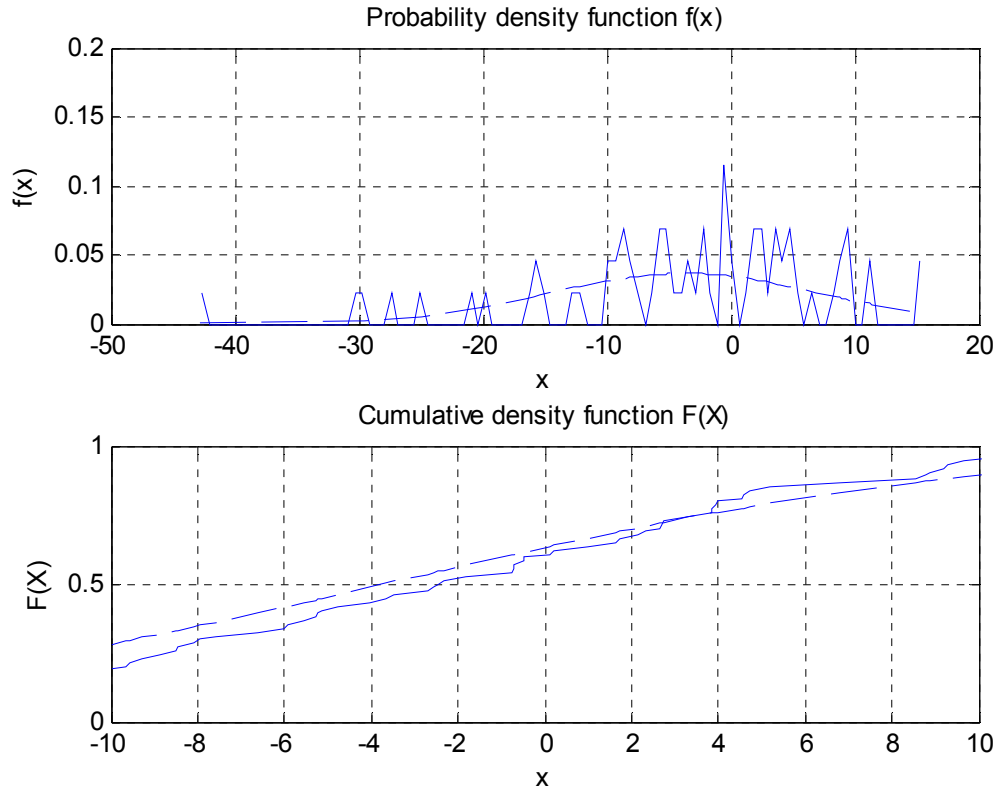


Figure 31. Weapon A - SN_CDF_data for deflection

Knowing that the data passes the confidence in fit to a single normal distribution in range and deflection allows the calculation of a single REP and DEP for entry into JWS. For this case, REP = 13.358 and DEP = 7.362.

B. CASE STUDY 2

1. Delivery Accuracy

The data found for this study can be found in Table 4. We begin by running the range and deflection data through GE_extractXY and find that no gross errors are removed ($P_{GE} = 0$). Now, we run the range and deflection vectors that are output from GE_extractXY through DN_CDF_data separately. Table 5 shows the statistical results and

Figures 32 and 33 show the plot of the test data, PDF, and CDF for range and deflection, respectively.

Weapon B						
Range miss distance	Deflection miss distance	radial miss distance		Range miss distance	Deflection miss distance	radial miss distance
0.3	-0.9	1		4.3	7.3	8.5
-8.8	-3.2	9.3		52.1	59.4	79
-5.7	-1.4	5.9		100.5	84.1	131.1
1.7	-1.6	2.4		72.4	82.4	109.7
0.1	1.8	1.8		-0.8	-3.3	3.4
-3.6	2.6	4.5		67.8	91.9	114.3
-1.6	-4.2	4.5		6.8	4.4	8.1
7.8	4.7	9.1		-4.6	5.7	7.3
-2.2	2.2	3.2		7.7	-13	15.1
-8.1	-0.1	8.1		84.3	33.6	90.7
-0.3	-1	1		72.9	37.5	82
2	-0.9	2.2		-19	-17.1	25.6
-1.6	6.1	6.3		-4.7	-18	18.6
-2.7	5.2	5.9		-10.4	-11.6	15.6
-2.3	4.3	4.9		-4.9	-14.8	15.6

Table 4. Weapon B data

DN_CDF_data								
Range					Deflection			
p	0.66				p	0.661		
$\mu 1$	-2.044	$\sigma 1$	3.683		$\mu 1$	0.588	$\sigma 1$	4.68
$\mu 2$	36.416	$\sigma 2$	53.299		$\mu 2$	18.692	$\sigma 2$	59.531

Table 5. Weapon B - DN_CDF_data results

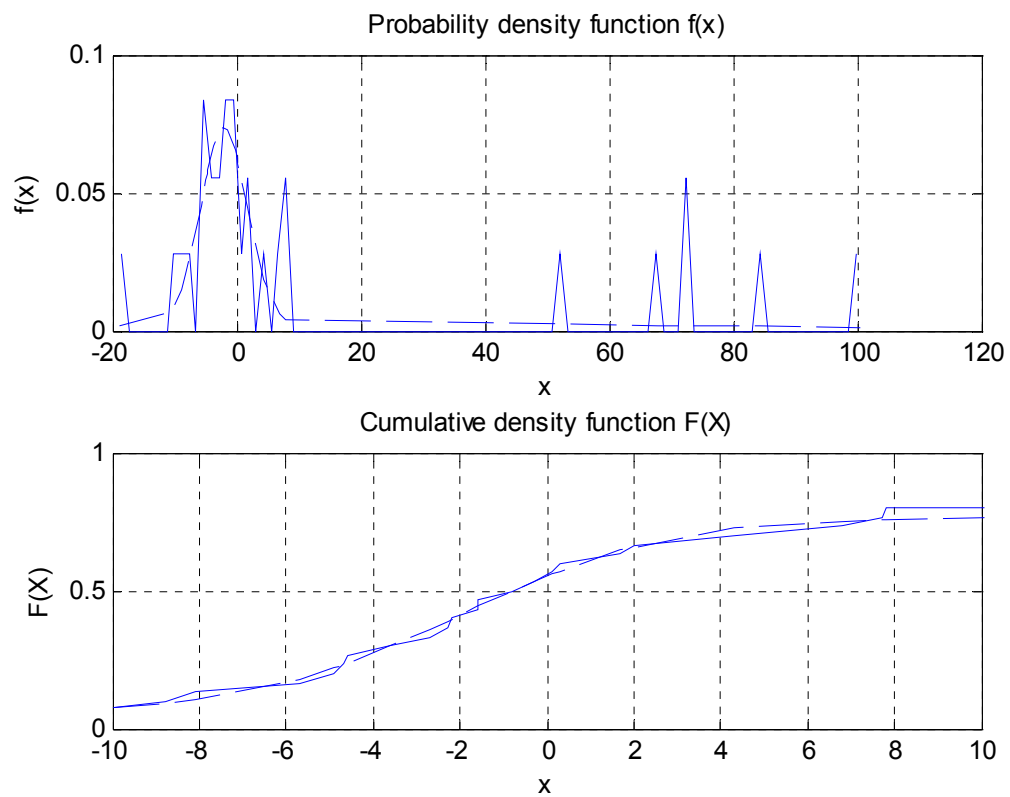


Figure 32. Weapon B - DN_CDF_data for range

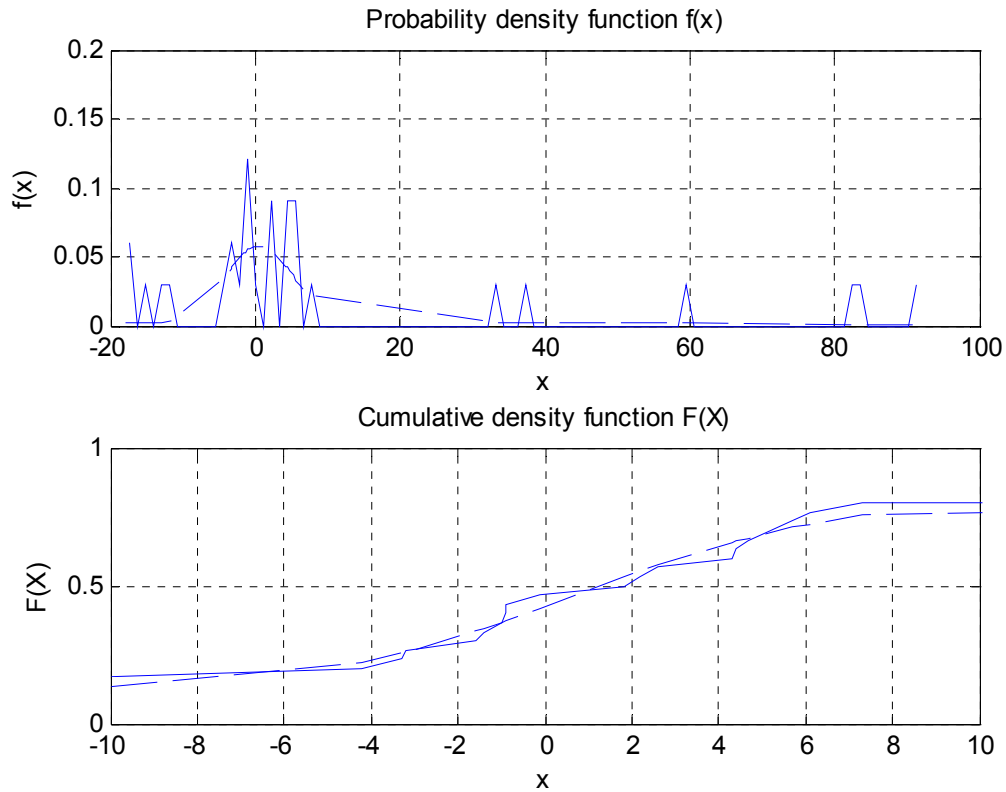


Figure 33. Weapon B - DN_CDF_data for deflection

From the analysis above, both distributions pass the confidence of fit and both are modeled well by double normal distributions. For range, we see a majority of the impacts are very close to the target but the remaining miss distance have a very large dispersion. For deflection impacts, we see the same statistical characteristics as in range. Clearly, a majority of misses are very close to the target, but those inaccurate impacts outside of this small boundary miss with a very large dispersion. Therefore, the data supplied to JWS for the probability of kill calculation is listed in Table 5.

2. Current Methodology

As JWS cannot currently accept double normal statistics, we will continue to evaluate the data set for a single normal distribution next. As a reminder there were no gross errors in the data set. In this case, both range and deflection miss distances fail the confidence of fit to a single normal distribution. Thus, we convert the data in Table 4 to radial miss distances and run the single vector through the DR_CDF_data program. The results are shown in Table 6 and the plot of the radial data, PDF, and CDF are shown in Figure 34.

DR_CDF_data	
Radial	
p	0.567
σ_1	3.86
σ_2	18.525

Table 6. Weapon B - DR_CDF_data results

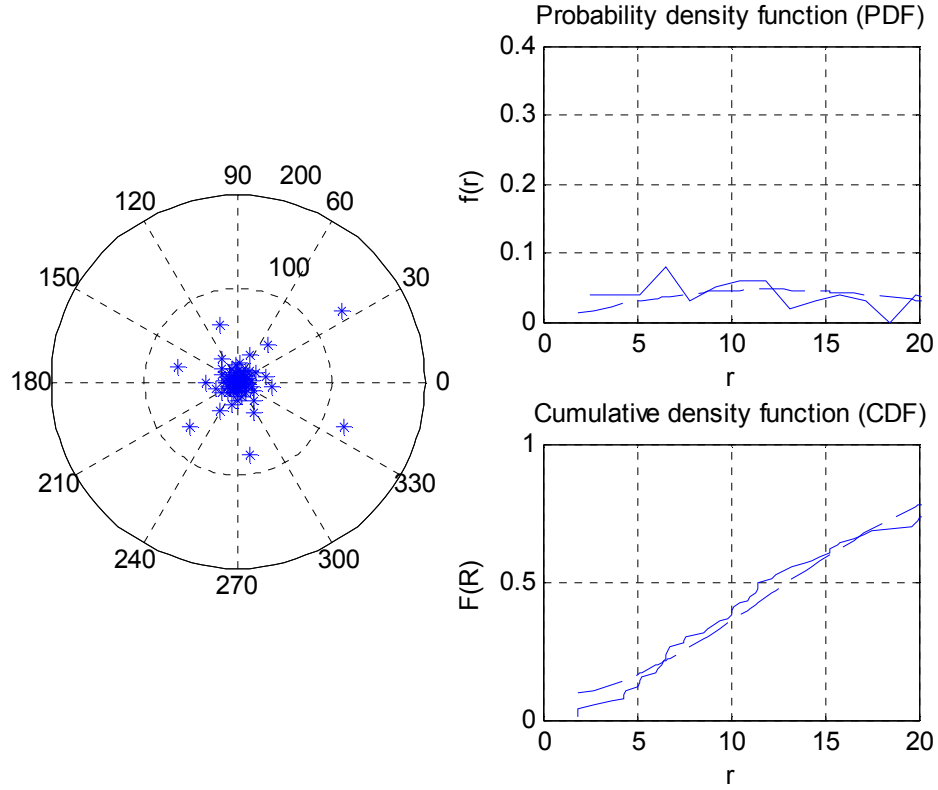


Figure 34. Weapon B - DR_CDF_data for radial misses

In this case, the data passes the confidence of fit to a double Rayleigh distribution. We can calculate two values of CEP and input them into JWS. In this case, $CEP_1=4.544$ and $CEP_2= 21.811$. JWS will output a probability of kill values for each CEP separately and we find the total probability of kill by combining the P_k values according to the weighting factor p as in Equation (24), repeated here for convenience.

$$Pk_T = Pk_1 * p + Pk_2 * (1 - p)$$

C. CASE STUDY 3

1. Delivery Accuracy

The data considered in this study is tabulated in Table 7. Running the data through GE_extractXY results in the removal of no gross errors ($P_{GE} = 0$). Next, we input the range and deflection vectors into the DN_CDF_data program separately and the results are displayed in Table 8.

Weapon C		
Range miss distance	Deflection miss distance	Radial miss distance
0.12	-2.66	2.662705
0.41	-0.23	0.470106
0.17	-0.59	0.614003
-1.07	-1.81	2.102617
-0.27	-0.24	0.361248
-1.52	-2.09	2.584279
-5.2	-14.54	15.44188
-0.54	-1.64	1.726615
1.82	-1.98	2.689387
0.99	-1.29	1.6261
1.28	-0.91	1.570509
1.82	-0.79	1.984061
-1.2	-2.96	3.193994
-1.02	-1.45	1.772823
-0.12	-0.32	0.34176
-0.12	-0.4	0.417612
3.46	-1.23	3.672125

Table 7. Weapon C data

DN_CDF_data							
Range					Deflection		
p	0.925				p	-1.116	
μ_1	-0.127	σ_1	1.173		μ_1	0.008	σ_1 0.044
μ_2	-0.442	σ_2	307.536		μ_2	0.001	σ_2 1.179

Table 8. Weapon C - DN_CDF_data results

For miss distances in range, the weighting factor is close to unity and the difference in dispersion is two magnitudes different. For the deflection data, the weighting factor is negative, which is a clear sign that a fit to a double normal distribution is not valid and the analyst has to interpret the results with care and judgment. The K-S test in each of these cases passes at a 95 percent confidence level and again the code is designed to always produce a solution. Clearly, the solution in this case is not valid. As neither data set fits to a double normal distribution, we run the range and deflection miss distances through SN_CDF_data separately. The results are summarized in Table 9.

SN_CDF_data							
Range					Deflection		
μ	-0.058	σ	1.85		μ	-1.287	σ 0.858
REP			1.247899		DEP		0.578721

Table 9. Weapon C - SN_CDF_data results

Both the range and deflection miss distance pass the confidence of fit test for a single normal distribution and the values returned are sensible. Thus, we can model the accuracy of data in both range and deflection with the descriptive statistics in Table 9. The values supplied to JWS would be REP= 1.25 and DEP=0.58.

X. THE TOOLBOX AND GPS GUIDED WEAPONS

In an effort to further analyze a specific subset of data, this chapter will focus on weapons that are guided by the Global Positioning System (GPS). Utilizing the toolbox, test data from GPS guided weapons fit double normal distributions in range and deflection with high confidence. In contrast, the JWS calculation uses a drastically different predictive method to determine the accuracy of the GPS guided munitions that results in a single distribution. This chapter will investigate and determine a resolution to this apparent paradox.

A. THE TOOLBOX AND GPS GUIDED DATA

Consider a sample of miss distances from a GPS guided weapon. The sample consists of 289 test data points. We implement the toolbox methodology and remove 14 gross errors and find that the result fits a double normal distribution on range and deflection at 95 percent confidence. The resulting statics are shown in Table 10 and the plot of the impact points is shown in Figure 35.

DN_CDF_data								
Range					Deflection			
p	0.457				p	0.728		
μ_1	1.163	σ_1	12.028		μ_1	-0.166	σ_1	5.415
μ_2	-1.444	σ_2	3.246		μ_2	-0.126	σ_2	14.835

Table 10. DN_CDF_data results

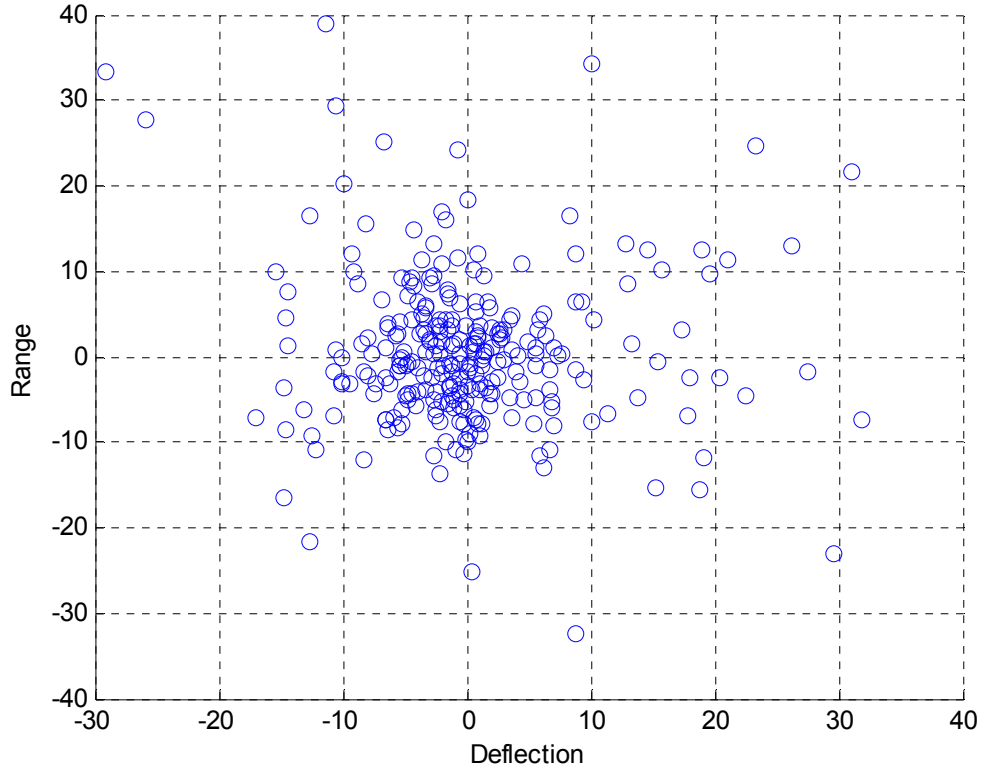


Figure 35. Test data impact points

Knowing this result, using the toolbox methodology we would provide JWS the statistics from Table 10 and allow it to calculate a probability of kill. However, this is not the methodology implemented by JWS.

B. GPS BACKGROUND

It should not be surprising that the GPS system cannot guide a weapon to an exact point on the ground. There are errors associated with system and they must be accounted for. The equation to calculate the total error for a GPS guided weapon is shown in Equation (25) [1].

$$Error_{TOTAL}^2 = (NAV)^2 + (G \& C)^2 + (TLE)^2 \quad (25)$$

The first error is the navigation error or NAV. This error is inherent to the GPS system. The sources of these errors are rooted in different system attributes and include: satellite clock error, ephemeris, troposphere, ionosphere, noise, and multipath. More specific definitions of these errors can be found in [6]. It is important to point out that these errors are presented as standard deviations and typical values of the errors are shown in Table 11 [1].

Error Source	GPS Ranging Error (m)	Error Type
Satellite clock	0.4	Bias
Orbit - ephemeris	0.4	Bias
Troposphere	0.5	Bias
Ionosphere	0.5	Bias
Receiver noise	0.4	Random
Multi-path	0.5	Random
Total RMS error	1.11	

Table 11. Standard GPS ranging errors

Further, these errors can be separated into two subgroups: random and bias. Satellite clock, ephemeris, troposphere, and ionosphere are classified as bias errors. Noise and multipath are considered random errors. Each facet that introduces error is given a certain accuracy value based on a unit length, which is a function of the quality of the GPS signal. The total error is found by root sum squaring (RSS) each individual error.

The guidance and control error (G&C) is next and is a function of the weapons position control system. The ability of the weapon to track the exact GPS location and use its control surfaces to maintain the error as close to

zero is the basis of this error. The error is nested in the weapon design and can be represented by a single constant standard deviation for a given weapon variant.

The last error is called the target location error (TLE). This error comes from the source of the GPS coordinates of the target and is also represented by a standard deviation. If using a map to identify a target location, this error could be significant. Using a spotter on the ground would reduce this error. If the test data comes from an instrumented range, this error is assumed to be zero.

The only error that varies when conducting tests on an instrumented range is the NAV error. The guidance and control error is constant and known based on the weapon that we are testing. The TLE error is assumed zero as we know the exact location of the target. These errors are the first step in generating the GPS Weapons Delivery Accuracy Program (DWDAP) shown in Figure 36.

GPS Weapon Accuracy Calculator - (inputs are green cells) Distances in meters						
	Bias	Random				
Satellite clock	0.780					
Ephemeris	0.780			HDOP	1.1	
Troposphere	0.500			VDOP	1.7	
Ionosphere	1.000			GDOP	2.02	
Noise		0.400		Impact θ	60	
Multipath		0.500				
Receiver	1.288					
Space and clock (S&C)	1.103					
RSS	1.571	0.640				
UERE (pseudo range sigma)	1.696					
	Horizontal			Vertical		
	Bias	Random	Total H	Bias	Random	Total V
Nav error H (sigma)	1.222	0.498	1.319	2.670	1.089	2.883
G&C H (sigma)		0.590	0.590		0.590	0.590
TLE H (sigma)	2.840		2.840	3.710		3.710
RSS (sigma)	3.092	0.772	3.187	4.571	1.238	4.736
	Range			Deflection		
	Bias	Random	Total R	Bias	Random	Total D
Summary-horizontal plane	4.065	1.052	4.199	3.092	0.772	3.187
REP/DEP H plane						
Single weapon CEP H plane			4.326			
	Vertical			Deflection		
	Bias	Random	Total V	Bias	Random	Total D
Summary-vertical plane	7.040	1.238	7.148	3.092	0.772	3.187
HEP/DEP V plane						
Single weapon CEP V plane			5.984			

Figure 36. GPS Weapon Accuracy Calculator

C. NAV ERROR

The GPS ranging errors are combined by root sum squaring (RSS) each individual error from Table 11 to arrive at the user equivalent range error (UERE). The UERE is then modified by 3 parameters: the horizontal dilution of precision (HDOP), the vertical dilution of precision (VDOP) and impact angle. The dilution of precision is a function of the GPS satellite locations relative to the

receiver [6]. The larger the bearing spread in the satellites that you are receiving, the lower the dilution of precision. Similarly, if the satellites do not have a lot of bearing spread, the value of the dilution of precision is higher. The impact angle is a function of the height and distance from which a payload is dropped. This may be controlled by the mission planner and can be entered for each launch separately.

The dilution of precision errors, HDOP and VDOP, are used to correct the navigation error in the horizontal and vertical planes, respectively. The range and deflection errors are then calculated using the horizontal and vertical errors and corrected by the impact angle. This produces a single REP and DEP, which can be used separately or combined into a CEP value in the horizontal plane. The same procedure can be used in the vertical plane if this is the frame that analysis is being performed. Thus, given a weapon variant on an instrumented range, the only input that changes in the calculator is HDOP, VDOP and the impact angle.

The process of combining these errors and producing REP and DEP or CEP is implemented in the GPS Weapons Accuracy Calculator shown in Figure 36 [1] where the green cells are inputs.

D. RESOLUTION

The calculator in Figure 36 is the methodology used by JWS when calculating the accuracy of GPS guided weapons. It is imperative that we point out a fundamental feature in the derivation of the GPS Weapons Accuracy Calculator. When combining distributions that are represented by standard

deviations or individual Gaussian errors, the resulting distribution is a different single normal distribution. Thus, each shot can be characterized by a single normal distribution and the GPS Weapons Accuracy Calculator evaluates the weapon accuracy on a shot to shot basis.

In contrast, the toolbox that was presented in earlier chapters does not consider each shot on an individual basis. The toolbox considered all shots as a sample and fits a double normal distribution to 275 occurrences. We now investigate whether these two approaches can be reconciled.

E. GPS GUIDED WEAPON CASE STUDY

As shown earlier in this chapter, we fit a double normal distribution in range and deflection to a sample of 275 test impacts for a GPS guided weapon after 14 gross errors were removed. The statistics are shown in Table 10. However, our argument above using the GPS calculator is that each occurrence should be described by a single normal distribution in range and deflection, respectively. Thus, we must resolve the dilemma of whether the test data should be represented as 275 single normal distributions in range and deflection or if the data is one double normal distribution with 275 data points for range and deflection.

1. HDOP and VDOP

Many parameters like impact angle are recorded during a testing events but the value of HDOP and VDOP at the time of the test was not provided with the data. As the GPS Accuracy Calculator requires this as an input, the Federal Aviation Administration (FAA) provided a sample of HDOP and

VDOP for approximately the same geographic location as the testing range. The data for HDOP and VDOP are plotted over the course of 30 days as shown in Figure 36 and Figure 37. From the plots, ignoring the large singular peaks that only last thirty minutes for the entire month, the upper and lower bounds on HDOP are 1.3 and 0.6, respectively. Considering VDOP, ignoring the singular peaks, the upper and lower bounds are 2.1 and 0.9, respectively.

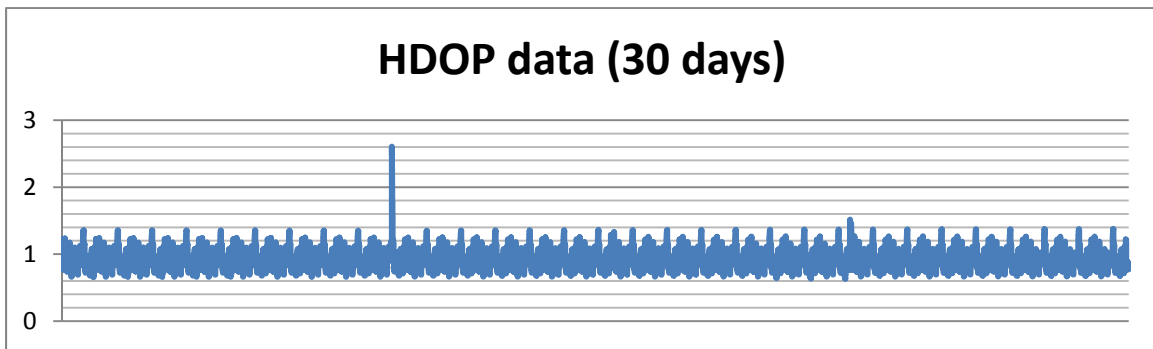


Figure 37. HDOP plotted for 30 days

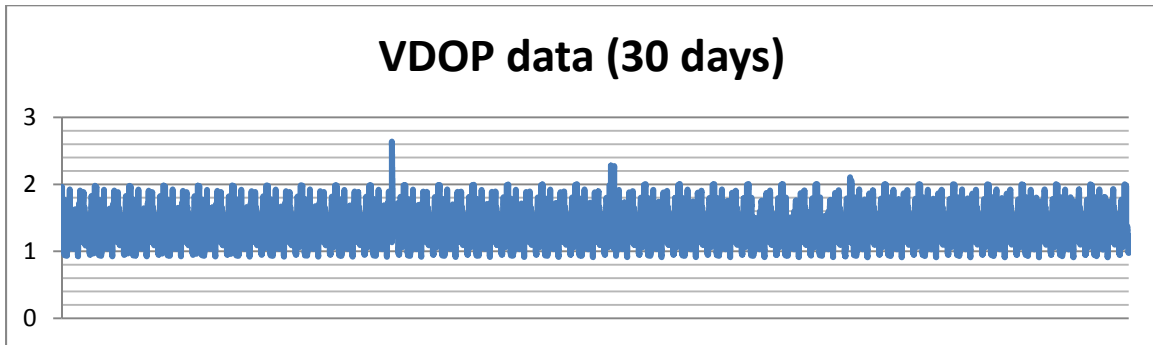


Figure 38. VDOP plotted for 30 days

2. HDOP/VDOP and the Test Data

Given the HDOP and VDOP bounds, we generate a data set using the lower HDOP bound and lower VDOP bound with the same impact angles that were recorded during the testing.

This data set was generated using the MATLAB script `GPS_accuracy_calc` found in Appendix B. Similarly, we generate a second data set using the upper HDOP bound and upper VDOP bound, again using the same set of impact angles as those recorded during the test. The plot of the actual test data and the two data sets generated from our HDOP/VDOP bounds is shown in Figure 38.

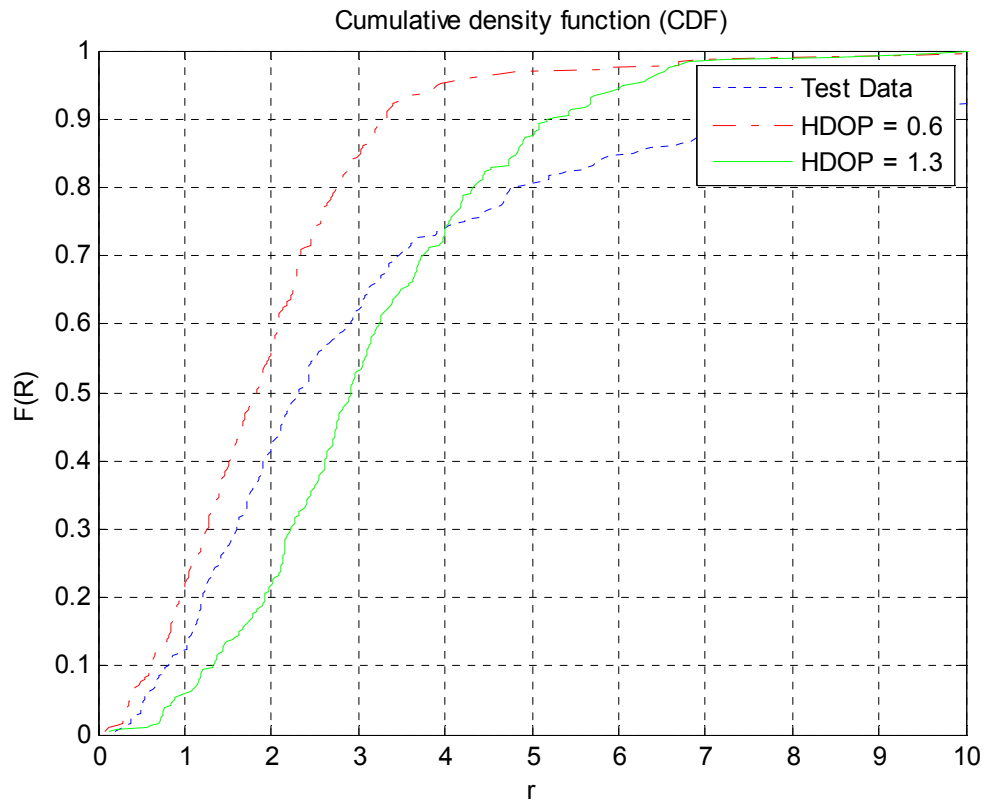


Figure 39. Data vs. HDOP/VDOP (same impact angle)

As seen in the plot, we are able to place bounds on the data set up to the 72th percentile.

3. Data Refinement

In an effort to improve our modeling, it is necessary to look at the data set with more scrutiny. First, gross errors must be removed from the data set. This is accomplished using the GE_extractXY program from the toolbox. This is reason that the actual test data has such a large tail in Figure 38. Next, in our generating data set, we assumed the TLE was zero as the data was recorded on an instrumented range. Upon further inspection, some of the tests used pilot designated targets or air-to-ground radar to enter target coordinates into the weapon. It is also required that we remove these data points as it is contrary to our assumptions of zero TLE. Removing these data points results in a remaining sample of 242 occurrences.

4. Modeling Impact Angle

In Figure 38, we used the same set of impact angles that were recorded in the data set so that we were only changing one variable at a time. Since we know that we can bound that data, it is desirable to more generally model the set of impact angles for comparison. A plot of the distribution of impact angles can be found in Figure 39.

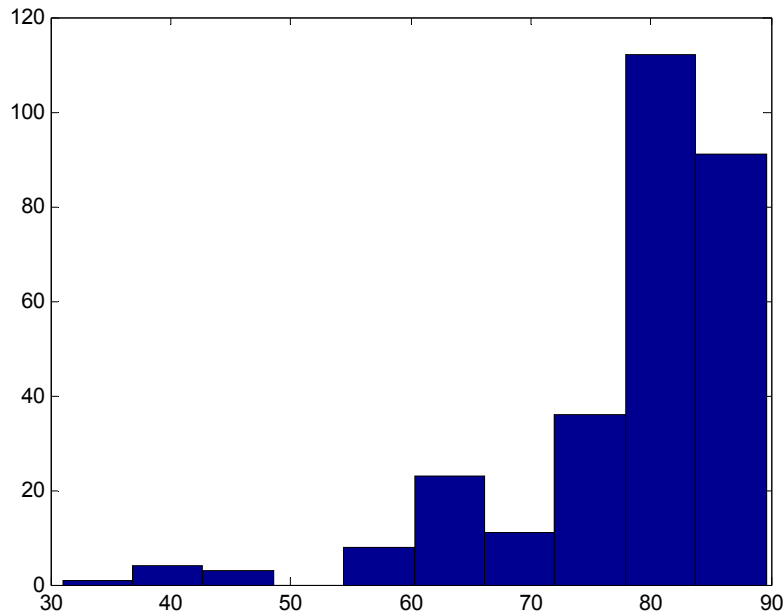


Figure 40. Histogram of Impact Angles

We see that the impact angles are random and it is imperative that we model these as accurately as possible. For this data set, we have chosen to model the impact angles as a single normal distribution with a mean of 78 degrees and a standard deviation of 3 degrees. In effect, this limits the range of impact angle inputs to the GPS calculator from 66 to 90 degrees. This also reduces the models dependence on VDOP, as the angular correction at such a steep impact angle is very small. This generalization of impact angles is also consistent with tactics usually employed for such weapons [3].

5. Updated Model

Removing the gross errors and the events that have a TLE not equal to zero, as well as generalizing the impact angles, results in the updated model in Figure 40.

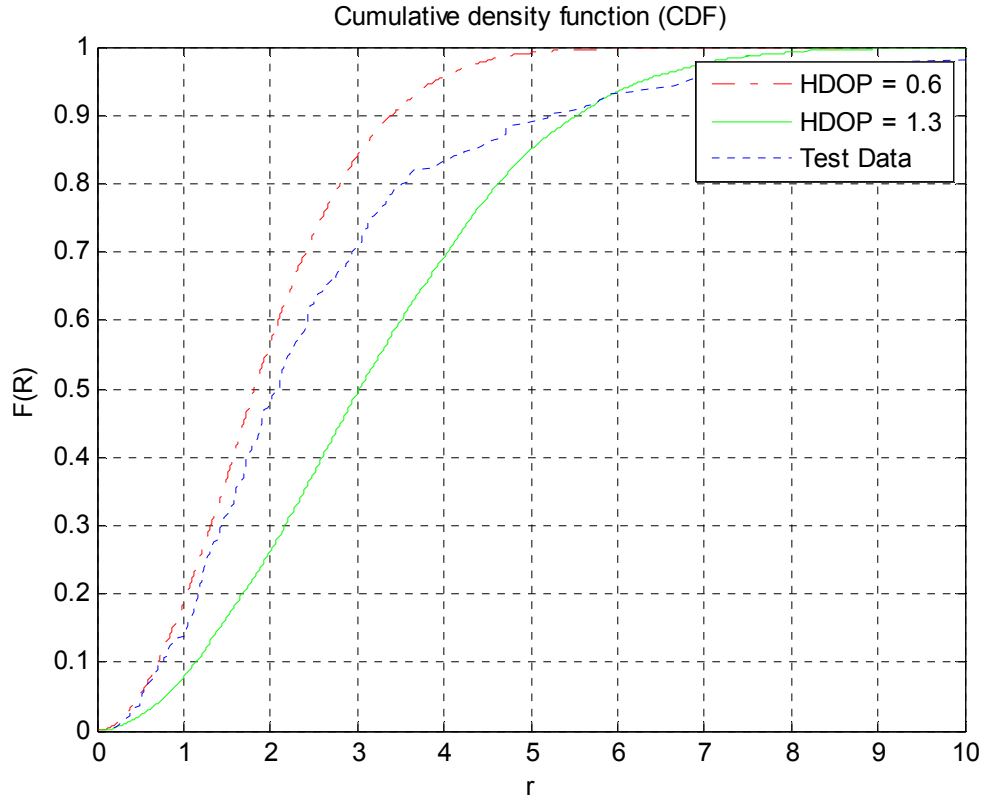


Figure 41. Data vs. HDOP/VDOP (generalized impact angle)

In this case, we compared the test data to a sample of ten thousand random draws varying the impact angle as described earlier. Additionally, the upper and lower bound of HDOP from the FAA are used again but the data is no longer dependent on VDOP as our impact angles are steep. Thus, VDOP was set at an average value of 2.1. From Figure 40, we see that the updated model bounds 93% of the data. It is also likely that the 7% of data that is outside the bound is from shallow impact angles and has a strong dependence on VDOP. This dependence was removed from the generalization, as mission planners do not plan for shallow impact angles. This bound has extreme importance, as we should not model GPS guided munitions using aggregate data

sets of large sample sizes (242 different single normal distributions). Rather, each weapons firing can be described by a unique single normal distribution and given the HDOP, VDOP and the impact angle, we can describe the exact single normal distribution for that event. The REP and DEP from this distribution are used to calculate a probability of kill for each individual shot.

To emphasize, mission planners can immediately calculate REP, DEP and CEP values given values of HDOP, VDOP and impact angle for the mission plan using the spreadsheet in Figure 36. Depending on the value of the target and the importance for a direct hit, the values of HDOP or VDOP could be used from historical archives, real-time readings or predictive estimates.

THIS PAGE INTENTIONALLY LEFT BLANK

XI. CONCLUSIONS AND RECOMMENDATIONS

The toolbox presented creates a dynamic and functional solution to the standardization of weapons accuracy. The following conclusions were drawn while developing the toolbox.

1. The toolbox developed characterizes the accuracy of several weapons systems. For all the test data files that were analyzed, the toolbox resolved each data set to fit either a single normal distribution, double normal distribution, single Rayleigh distribution, or a double Rayleigh distribution with 95 percent confidence.

2. The toolbox fitted a double normal distribution to several sets of data for GPS guided weapons, however JWS would consider the same coordinate seeking weapon to have a single normal distribution as predicted by the GWDAP (Figure 36) calculator. The correct interpretation of N independent test data points should therefore be a collection of N single normal distributions, where N represents the number of independent test firings. Test data from coordinate seeking weapons should continue to be modeled as a single normal distribution based on each independent weapon firing as is currently done by GWDAP.

3. With respect to other guided weapons systems (e.g., laser guided bombs); the physical basis of weapon accuracy should be investigated for the same premise as in number 2 above. Essentially, one could perhaps develop a physics based mathematical model that represents laser guided bomb error as a single normal distribution, whereas currently

the accuracy of such weapons is calculated as a distribution from a set of N independent test data points.

APPENDIX

A. MATLAB CODE - DN_CDF_DATA

% This program reads linear miss distance data (range or deflection)
% and then fits a double normal distribution in order to determine the
% distribution stats. The resulting PDF, CDF and data are plotted

```
clc, clf, clear all
```

```
[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to  
evaluate');  
ExPath = [PathEx FileEx];
```

```
handles.ExPath = ExPath;
```

```
diary DN_data.txt;  
%----- Read in all the linear miss distance data -----  
%s=input('Type in the name of the file to be analyzed (incl. path and  
%extension): \n\n', 's');  
%s='Data sets/JDAM_LAT_range.txt';  
%s='Data sets/test4.txt';  
s=ExPath;  
fid=fopen(s); % open the data file  
x=fscanf(fid,'%f',[1,inf]); % read entire data file  
ST=fclose(fid); % close datafile  
s_init=length(x); % number of samples  
mu1=mean(x); % initial stats  
s1=std(x);  
fprintf('Before outliers are removed from %4i samples\n', s_init)  
fprintf('mean = %4.3f m, sigma= %4.3f m\n',mu1,s1)  
%----- Remove Gross Errors -----  
s_init=length(x);  
for i=s_init:-1:1 % count down since vector gets shorter  
    if( abs(x(i))>4*s1 )  
        x(i)=[];  
    end  
end  
s_final=length(x);  
mu2=mean(x); % calc new stats  
s2=std(x);  
Pge=(s_init-s_final)/s_init;  
fprintf('\nAfter %2i outliers are removed\n',s_init-s_final)  
fprintf('mean = %4.3f m, sigma= %4.3f m\n',mu2,s2)  
%----- Calc CDF -----  
x=sort(x);  
for i=1:s_final  
    CDF_data(i)=i/s_final;  
end  
%- -----Call FMINSEARCH to fit the data -----  
Starting=[0.5 0.0 0.0 1.0 5.0]; % w1, mu1, mu2, s1, s2  
options=optimset('MaxFunEvals',5000);
```

```

Estimates=fminsearch(@DN_myfit,Starting,options,x,CDF_data);

% ----- Predict CDF data using estimated parameters -----
w1e=Estimates(1);
m1e=Estimates(2);
mu2e=Estimates(3);
s1e=Estimates(4);
s2e=Estimates(5);
cdf1=cdf('Normal',x,m1e,s1e);
cdf2=cdf('Normal',x,mu2e,s2e);

CDF_pred=w1e*cdf1+(1-w1e)*cdf2;
fprintf('\nCalling curve fit to the CDF data...\n')
fprintf('weighting factor = %4.3f\n',w1e)
fprintf('mean1 = %4.3f m, sigma1= %4.3f m\n',m1e,s1e)
fprintf('mean2 = %4.3f m, sigma2= %4.3f m\n',mu2e,s2e)

% ----- Plot the CDF, data and predicted -----
subplot(2,1,2);
plot(x,CDF_data);           % plot data
hold on;
xlim=10;
axis([-xlim xlim 0 1]);
subplot(2,1,2)
plot(x,CDF_pred,'--');      % plot predicted
xlabel('x')
ylabel('F(X)')
title('Cumulative density function F(X)')
grid on;
hold off;

%----- Plot the histogram of data-----
nbins = 100;
[h, centres] = hist(x, nbins);
% normalise to unit area
norm_h = h / (numel(x) * (centres(2)-centres(1)));
subplot(2,1,1)
plot(centres, norm_h);
hold on;

% -----compute and display double normal PDF -----
pdf1=pdf('Normal',x,m1e,s1e);
pdf2=pdf('Normal',x,mu2e,s2e);
PDF_pred=w1e*pdf1+(1-w1e)*pdf2;
plot(x,PDF_pred,'--');      % plot data
hold on;
xlabel('x')                 % label axes
ylabel('f(x)')
title('Probability density function f(x)')
grid on
hold off;

%----- KS test -----
CDF_test=[CDF_data',CDF_pred']; % form the CDF vector for the KS
test
for conf_level=95:-5:60
    [h,p,ksstat,cv]=kstest(CDF_data',CDF_test,1-conf_level/100);

```

```

        if h==1
            fprintf('\nTest for CDF fit FAILS at the %2i percent confidence
level',conf_level)
        else
            fprintf('\nTest for CDF fit PASSES at the %2i percent
confidence level',conf_level)
            break
        end
    end
end
fprintf('\n')
diary off

```

1. MATLAB CODE DN_myfit

```

function sse=myfit(params,Input,Actual_Output)
wlf=params(1);
mulf=params(2);
mu2f=params(3);
s1f=params(4);
s2f=params(5);
cdf1=cdf('Normal',Input,mulf,s1f);
cdf2=cdf('Normal',Input,mu2f,s2f);

Fitted_Curve=wlf*cdf1+(1-wlf)*cdf2;

Error_Vector=Fitted_Curve - Actual_Output;

% Minimize the sum of squares error
sse=sum(Error_Vector.^2);

```

B. MATLAB CODE - DR_CDF_DATA

```

% This program reads radial miss distance data and
% then fits a double Rayleigh distribution in order to determine the
% distribution stats. The resulting PDF, CDF and data are plotted

clc, clf, clear all

[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];
handles.ExPath = ExPath;

diary DR_data.txt;
%----- Read in all the radial miss distance data -----
s=ExPath;
fid=fopen(s); % open the data file
x=fscanf(fid,'%f',[1,inf]); % read entire data file
ST=fclose(fid); % close data file
s_init=length(x); % number of samples
mul=mean(x); % initial stats
s1=std(x)/.655;
median_r=median(x);
x=sort(x);

```

```

fprintf('Before outliers are removed from %4i samples\n', s_init)
fprintf('mean = %4.3f m, sigma = %4.3f m, median = %4.3f\n', mu1, s1, median_r)
%----- Remove Gross Errors -----
s_init=length(x);
for i=s_init:-1:1 % count down since vector gets shorter
    if( abs(x(i))>4*s1 )
        x(i)=[];
    end
end
s_final=length(x);
mu2=mean(x); % calc new stats
s2=std(x)/.655;
median_r=median(x);
Pge=(s_init-s_final)/s_init;
fprintf('\nAfter %2i outliers are removed\n',s_init-s_final)
fprintf('mean = %4.3f m, sigma = %4.3f m, median = %4.3f\n', mu2, s2, median_r)
%----- Calc CDF_data -----
for i=1:s_final
    CDF_data(i)=i/s_final;
end
%-----Call FMINSEARCH to fit the data -----
fprintf('\nCalling curve fit to the CDF data...\n')
Starting=[0.5 1.0 5.0]; % initial gess for w1, s1, s2
options=optimset('MaxFunEvals',5000);
Estimates=fminsearch(@DR_myfit,Starting,options,x,CDF_data);
% ----- Predict CDF data using estimated parameters -----
w1e=Estimates(1);
s1e=Estimates(2);
s2e=Estimates(3);
cdf1=raylcdf(x,s1e);
cdf2=raylcdf(x,s2e);
CDF_pred=w1e*cdf1+(1-w1e)*cdf2;
fprintf('weighting factor = %4.3f\n',w1e)
fprintf('sigma1= %4.3f m\n',s1e)
fprintf('sigma2= %4.3f m\n',s2e)
% ----- Plot the CDF, data and predicted -----
subplot(2,1,2);
plot(x,CDF_data); % plot data
hold on;
%plot(x,cdf1,'g')
%plot(x,cdf2,'g')

xlimit=20;
subplot(2,1,2)
plot(x,CDF_pred,'r'); % plot predicted
axis([0 xlimit 0 1.0]);

xlabel('r')
ylabel('F(R)')
title('Cumulative density function (CDF)')
grid on;
hold off;

```

```

%----- Plot the histogram of data-----
nbins = 100;
[h, centres] = hist(x, nbins);
% normalise to unit area
norm_h = h / (numel(x) * (centres(2)-centres(1)));
subplot(2,1,1)
plot(centres, norm_h);
axis([0 xlimit 0 0.4]);
hold on;
% -----compute and display double Rayleigh PDF -----
pdf1=x./(s1e*s1e).*exp(-x.*x/(2*s1e*s1e));
pdf2=x./(s2e*s2e).*exp(-x.*x/(2*s2e*s2e));
PDF_pred=w1e*pdf1+(1-w1e)*pdf2;
axis([0 xlimit 0 0.4]);
plot(x,PDF_pred,'r');
hold on;
%plot(x,pdf1,'g')
%plot(x,pdf2,'g')
xlabel('r')
ylabel('f(r)')
title('Probability density function (PDF)')
grid on
hold off;
%----- KS test -----
CDF_test=[CDF_data',CDF_pred']; % form the CDF vector for the KS
test
% conf_level=0.95;
for conf_level=95:-5:60
    [h,p,ksstat,cv]=kstest(CDF_data',CDF_test,1-conf_level/100);
    if h==1
        fprintf('\nTest for CDF fit FAILS at the %2i percent confidence
level',conf_level)
    else
        fprintf('\nTest for CDF fit PASSES at the %2i percent
confidence level',conf_level)
        break
    end
end
fprintf('\n')
diary off

```

1. MATLAB CODE DR_myfit

```

function sse=myfit(params,Input,Actual_Output)
w1f=params(1);
s1f=params(2);
s2f=params(3);
Fitted_Curve=1-w1f.*exp(-Input.*Input/(2*s1f*s1f))-(1-w1f).*exp(-
Input.*Input/(2*s2f*s2f));
Error_Vector=Fitted_Curve - Actual_Output;
% When curvefitting, a typical quantity to
% minimize is the sum of squares error
sse=sum(Error_Vector.^2);

```

C. MATLAB CODE - GE_EXTRACT_XY

```
% This program reads X,Y miss distance data, extracts the gross errors
% and then writes a file of the remaining data for further analysis

clc, clear all
[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];
handles.ExPath = ExPath;

diary GE_extractXY.txt;

fprintf('\n\n-----
----- ')
fprintf('\n-----This program extracts gross errors from an input
file----- ')
fprintf('\n-----
-----')
%----- Read in all the X, Y miss distance data -----
s_in=ExPath;
fid=fopen(s_in); % open the data file
Temp =fscanf(fid,'%f',[2,inf]); % read entire data file
Temp = Temp';
x = Temp(:,1);
y = Temp(:,2);
x = x';
y = y';
ST=fclose(fid); % close datafile
initial_size = length(x);
%-----Calculate initial Non-parametric data for X -----
sx_init=length(x); % number of samples
mux1=mean(x); % initial stats
sx1=std(x);
sigma_x1=sx1/0.6551;
median_x=median(x);
len_data=sx_init;
fprintf('\n-----Non-parametric data for Range-----
-----')
fprintf('\nBefore outliers are removed from %4i samples, \n', sx_init);
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f
m\n',mux1,sx1,median_x);
%-----Calculate initial Non-parametric data for X -----
sy_init=length(y); % number of samples
muy1=mean(y); % initial stats
sy1=std(y);
sigma_y1=sy1/0.6551;
median_y=median(y);
len_data=sy_init;
fprintf('\n-----Non-parametric data for Deflection---
-----')
fprintf('\nBefore outliers are removed from %4i samples, \n', sy_init)
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f
m\n',muy1,sy1,median_y)
%----- Remove Gross Errors From X -----
```

```

j=1;
sum=0;
%sigma_l=s1;
while j<100

    s_init=length(x);
    num_sigma=4;
    for i=s_init:-1:1 % count down since vector gets shorter
        if( abs(x(i))>num_sigma*sigma_xl)
            x(i)=[];
            y(i)=[]; %remove the corresponding coordinate in y
        end
    end
    s_final=length(x);
    mu=mean(x); % calc new stats
    sigma_r=std(x);
    median_r=median(x);
    sigma_xl=sigma_r/0.655;
    Pge=(s_init-s_final)/s_init;
    x_max=max(x);
    delta=s_init-s_final;
    sum=sum+delta;
    if(delta<1)
        break
    end
    j=j+1;
    sum;
end
%fprintf('\n\n-----Interate over the Deflection data-----\n\n');
s_init=length(y); % number of samples
mu1=mean(y); % initial stats
s1=std(y);
sigma_l=s1/0.6551;
median_r=median(y);
len_data=s_init;
%----- Remove Gross Errors From Y -----
j=1;
sum=0;
%sigma_l=s1;
while j<100
    s_init=length(y);
    num_sigma=3;
    for i=s_init:-1:1 % count down since vector gets shorter
        if( abs(y(i))>num_sigma*sigma_y1)
            x(i)=[]; %remove the entries in both x and y vectors
            y(i)=[];
        end
    end
    s_final=length(y);
    mu=mean(y); % calc new stats
    sigma_r=std(y);
    median_r=median(y);
    sigma_y1=sigma_r/0.655;
    Pge=(s_init-s_final)/s_init;

```

```

        y_max=max(y);
        delta=s_init-s_final;
        sum=sum+delta;
        if(delta<1)
            break
        end
        j=j+1;
        sum;
    end
    %Final Statistics X
    sx_final=length(x);
    muxf=mean(x);
    sxf=std(x);
    sigma_xf=sxf/0.6551;
    median_xf=median(x);
    len_data=sx_final;
    %Final Statistics Y
    sy_final=length(y);
    muyf=mean(y);
    syf=std(y);
    sigma_yf=sy1/0.6551;
    median_y=median(y);
    len_data=sy_init;

    num_error = initial_size - length(x);

    fprintf('\n-----\n-----')
    fprintf('\n-----\n-----')
    fprintf('\n\n-----Final statistics-----\n\n')
    fprintf('Number of Gross errors removed = %4.3f, P_GE = %4.3f',
    num_error, num_error/initial_size)
    fprintf('\n\n-----Final data for Range-----\n-----')
    fprintf('\n mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f\n',muxf,sigma_xf,median_xf)
    fprintf('\n-----Final data for Deflection-----\n-----')
    fprintf('\n mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f\n\n',muyf,sigma_yf,median_y)
    A = [x;y];
    A = A';
    %-----write residual file-----
    s_out='GE_extract_output_file_range.txt';
    fid=fopen(s_out,'w');
    dlmwrite(s_out, x', 'delimiter', '\t','precision', 6)
    fclose(fid);
    s_out='GE_extract_output_file_deflection.txt';
    fid=fopen(s_out,'w');
    dlmwrite(s_out, y', 'delimiter', '\t','precision', 6)
    fclose(fid);
    diary off

```


D. MATLAB CODE - SN_CDF_DATA

```
% This program reads single normally distributed data and
% then fits a distribution to the CDF in order to recover the original
% distribution stats. The PDF and data are also plotted

clc, clf, clear all
[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];
handles.ExPath = ExPath;

diary SN_data.txt;

%----- Read in all the linear miss distance data -----
s=input('Type in the name of the file to be analyzed (incl. path and
extension): \n\n', 's');
s=ExPath;
fid=fopen(s); % open the data file
x=fscanf(fid,'%f',[1,inf]); % read entire data file
ST=fclose(fid); % close datafile
s_init=length(x); % number of samples
mu1=mean(x); % initial stats
s1=std(x);
fprintf('Before outliers are removed from %4i samples\n', s_init)
fprintf('mean = %4.3f m, sigma= %4.3f m\n',mu1,s1)
%----- Remove Gross Errors -----
s_init=length(x);
for i=s_init:-1:1 % count down since vector gets shorter
    if (abs(x(i))>4*s1 )
        x(i)=[];
    end
end
s_final=length(x);
mu2=mean(x); % calc new stats
s2=std(x);
Pge=(s_init-s_final)/s_init;
fprintf('\nAfter %2i outliers are removed\n',s_init-s_final)
fprintf('mean = %4.3f m, sigma= %4.3f m\n',mu2,s2)
%----- Calc CDF -----
x=sort(x);
for i=1:s_final
    miss(i)=x(i);
    CDF_data(i)=i/s_final;
end
% -----Plot the CDF data to be curve fitted -----
subplot(2,1,2);
plot(miss,CDF_data);
hold on;

limit=floor(s_final/2);
for i=1:s_final
    CDF_pred(i)=cdf('Normal',miss(i),mu2,s2);
end
```

```

xlimit=10;
axis([-xlimit xlimit 0 1]);
subplot(2,1,2)
plot(x,CDF_pred,'--');      % compare data with predicted function
title('Cumulative density function F(X)')
xlabel('x')
ylabel('F(X)')
grid on;
hold off;

CDF_test=[miss',CDF_pred'];      % form the CDF test vector for KStest

%----- Plot the PDF and data-----
% compute and display histogram of raw data with unity area using web
code
nbins = 100;
[h, centres] = hist(x, nbins);
% normalise to unit area
norm_h = h / (numel(x) * (centres(2)-centres(1)));
subplot(2,1,1)
plot(centres, norm_h);
hold on;
% compute and display normal PDF based on statistics of raw data
for i=1:s_final
    PDF_pred(i)=pdf('Normal',miss(i),mu2,s2);
end
plot(x,PDF_pred,'--');
hold on;
title('Probability density function f(x)')
xlabel('x')
ylabel('f(x)')
grid on;
hold off;

%----- KS test -----
CDF_test=[CDF_data',CDF_pred'];      % form the CDF vector for the KS
test
for conf_level=95:-5:60
    [h,p,ksstat,cv]=kstest(CDF_data',CDF_test,1-conf_level/100);
    if h==1
        fprintf('\nTest for CDF fit FAILS at the %2i percent confidence
level',conf_level)
    else
        fprintf('\nTest for CDF fit PASSES at the %2i percent
confidence level',conf_level)
        break
    end
end
fprintf('\n')
fprintf('\nError probable=%3f\n\n',s2*0.6745)

diary of

```

E. MATLAB CODE - SR_CDF_DATA

% This program reads radial miss distance data and

```

% then fits a single Rayleigh distribution in order to determine the
% distribution stats. The resulting PDF, CDF and data are plotted

clc, clf, clear all

[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];

handles.ExPath = ExPath;

diary SR_data.txt;

%----- Read in all the radial miss distance data -----
s=ExPath;
fid=fopen(s);                                % open the data file
x=fscanf(fid,'%f',[1,inf]);                  % read entire data file
ST=fclose(fid);                              % close datafile
s_init=length(x);                           % number of samples
mul=mean(x);                                % initial stats
s1=std(x)/.655;
median_r=median(x);
fprintf('-----Non-parametric data-----')
fprintf('\nBefore outliers are removed from %4i samples\n', s_init)
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f
m\n',mul,s1,median_r)

%----- Remove Gross Errors -----
s_init=length(x);
num_sigma=4;
for i=s_init:-1:1      % count down since vector gets shorter
    if( abs(x(i))>num_sigma*s1 )
        x(i)=[];
    end
end
s_final=length(x);
mu=mean(x);              % calc new stats
sigma_r=std(x)/.655;
median_r=median(x);
sigma_l=sigma_r;
Pge=(s_init-s_final)/s_init;
x_max=max(x);
x=sort(x);
fprintf('\nAfter %2i outliers are removed\n',s_init-s_final)
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f
m\n',mu,sigma_r,median_r)

% compute and display histogram of data with unity area using web code
nbins = 100;
[h, centres] = hist(x, nbins);
% normalise to unit area
norm_h = h / (numel(x) * (centres(2)-centres(1)));
subplot(2,1,1)
plot(centres, norm_h);

```

```

hold on;
% Now plot the theoretical Rayleigh PDF
PDF_pred=(x/(sigma_1*sigma_1)).*exp(-x.*x/(2*sigma_1*sigma_1));
plot(x,PDF_pred,'r')
title('Probability density function f(x)')
xlabel('x')
ylabel('f(x)')
grid
hold off

%----- Calc and plot CDF -----
for i=1:s_final
    CDF_data(i)=i/s_final;
end

CDF_pred=1-exp(-x.*x/(2*sigma_1*sigma_1)); % now the predicted
subplot(2,1,2);
plot(x,CDF_data);
hold on;
plot(x,CDF_pred,'r')
title('Cumulative density function F(X)')
xlabel('x')
ylabel('F(X)')
grid
hold off

%----- Extract the sigma_1 from the data-----
fprintf('-----Rayleigh fitted data-----')
[sigma_fit,PCI]=raylfit(x)
%fprintf('From fitted data: sigma_')

%----- KS test -----
CDF_test=[CDF_data',CDF_pred']; % form the CDF vector for the KS test
for conf_level=95:-5:60
    [h,p,ksstat,cv]=kstest(CDF_data',CDF_test,1-conf_level/100);
    if h==1
        fprintf('\nTest for CDF fit FAILS at the %2i percent confidence
level',conf_level)
    else
        fprintf('\nTest for CDF fit PASSES at the %2i percent
confidence level',conf_level)
        break
    end
end
fprintf('\n\n')

diary off;

```

F. MATLAB CODE - GE_EXTRACT

```

% This program reads radial miss distance data, extracts the gross
% errors and then writes a file of the remaining data for further
% analysis
clc, clear all

```

```

[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];
handles.ExPath = ExPath;

diary GE_extract.txt;

fprintf('\n\n-----\n')
fprintf('\n---This program extracts gross errors from an input file---\n')
fprintf('\n-----\n')

%----- Read in all the radial miss distance data -----
s_in=ExPath;
fid=fopen(s_in); % open the data file
x=fscanf(fid,'%f',[1,inf]); % read entire data file
ST=fclose(fid); % close datafile
s_init=length(x); % number of samples
mul=mean(x); % initial stats
s1=std(x);
sigma_l=s1/0.6551;
median_r=median(x);
len_data=s_init;
% fprintf('-----Non-parametric data-----\n')
fprintf('\nBefore outliers are removed from %4i samples, \n', s_init)
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f\n',mul,s1,median_r)
%----- Remove Gross Errors -----
j=1;
sum=0;
%sigma_l=s1;
while j<100
    s_init=length(x);
    num_sigma=3;
    for i=s_init:-1:1 % count down since vector gets shorter
        if( abs(x(i))>num_sigma*sigma_l)
            x(i)=[];
        end
    end
    s_final=length(x);
    mu=mean(x); % calc new stats
    sigma_r=std(x);
    median_r=median(x);
    sigma_l=sigma_r/0.655;
    Pge=(s_init-s_final)/s_init;
    x_max=max(x);
    x=sort(x);
    delta=s_init-s_final;
    sum=sum+delta;
    if(delta<1)
        break
    end
    fprintf('\nIteration #%2i, after %2i outliers are removed,\n',j,delta)

```

```

        fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f m, #GE
= %3i\n',mu,sigma_r,median_r, sum)
        j=j+1;
        sum;
end
fprintf('\n\n-----Final statistics----- \n')
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f m, P_GE =
%4.3f\n\n',mu,sigma_r,median_r, sum/len_data)

%-----write residual file-----
s_out='GE_extract_output_file.txt';
fid=fopen(s_out,'w');
fprintf(fid,'%5.2f\r\n',x);
fclose(fid);

diary off

```

G. MATLAB CODE - PHIT/PNM

```

% modified method for Phit and Pnm
% This script takes in 3 column vectors: 1- range, 2- deflections
% 3 - radial miss distances and returns Phit, Pnm, and CEP

clc, clf, clear all

[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];

handles.ExPath = ExPath;

diary PHIT_PNM.txt;

s_in=ExPath;
fid=fopen(s_in);
Temp =fscanf(fid,'%f',[3,inf]);
Temp = Temp'
Temp = sortrows(Temp,3);
Temp = flipud(Temp);
A = sort(Temp,3);
x = Temp(:,1);
y = Temp(:,2);
z = Temp(:,3);
x = x';
y = y';
z = z';
ST=fclose(fid);

begin_size =length(z);
num_error=0;
while j <100
x_avg = std(x);
y_avg = std(y);
sigma = (x_avg+y_avg)/2;

```

```

error = 4*sigma;

s_init=length(x);
for i=s_init:-1:1 % count down since vector gets shorter
    if( abs(z(i))>error)
        x(i)=[];
        y(i)=[];
        z(i)=[];
    end
end
s_final=length(x);
Pge=(s_init-s_final)/s_init;
x_max=max(x);
delta=s_init-s_final;
num_error=num_error+delta;
    if(delta<1)
        break
    end
    j=j+1;
    num_error;
end

MPI_x = mean(x);
MPI_y = mean(y);

num_hit = 0;
num_impact = begin_size-num_error;
g=0;
while g< 1000
    x_avg = std(x);
    y_avg = std(y);
    sigma = (x_avg+y_avg)/2
    bound = .5*sigma
    estimate = round(0.1175*length(z));
    size_z = length(z);
    count =0;
    for h=1:size_z
        if z(h)<= bound
            count = count+1;
        end
    end

    if count >= estimate
        temp = count- estimate;
        stop = size_z-temp+1;
        for p=size_z:-1:stop
            %for p=1:temp

                x(p)=[];
                y(p)=[];
                z(p)=[];
            end
        end

        if count <=estimate
            break;
        end
    end
end

```

```

        end
        g = g+1;
    end
    num_hit = num_impact-length(z);
    num_nm = begin_size-num_error-num_hit;
    CEP = 1.1774*sigma;

    fprintf('\n Sample size = %4.3f ', begin_size);
    fprintf('\n Nhit = %4.3f ', num_hit);
    fprintf('\n Nnm = %4.3f ', num_nm);
    fprintf('\n Nge = %4.3f ', num_error);
    fprintf('\n Phit = %4.3f ', num_hit/begin_size);
    fprintf('\n Pnm = %4.3f ', num_nm/begin_size);
    fprintf('\n Pge = %4.3f ', num_error/begin_size);
    fprintf('\n CEP = %4.3f ', CEP);
    fprintf('\n');
    diary off

```

H. MATLAB CODE - GE_EXTRACT_GAUSSIAN

```

% This program reads miss distance data, extracts the gross errors
% and then writes a file of the remaining data for further analysis
clc, clear all

[FileEx,PathEx] = uigetfile('*.txt','Select the Data File to
evaluate');
ExPath = [PathEx FileEx];
handles.ExPath = ExPath;

diary GE_extract.txt;

fprintf('\n\n-----
----- ')
fprintf('\n-----This program extracts gross errors from an input
file----- ')
fprintf('\n-----
-----')

%----- Read in all the radial miss distance data -----
----
s_in=ExPath;
fid=fopen(s_in);
x=fscanf(fid,'%f',[1,inf]);
ST=fclose(fid);
s_init=length(x);
mu1=mean(x);
s1=std(x);
sigma_1=s1/0.6551;
median_r=median(x);
len_data=s_init;
% fprintf('-----Non-parametric data-----
----')
fprintf('\nBefore outliers are removed from %4i samples, \n', s_init)

```



```

fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f
m\n', mu_l, s_l, median_r)
%----- Remove Gross Errors -----
----
j=1;
sum=0;
%sigma_l=s_l;
while j<100
    s_init=length(x);
    num_sigma=3;
    for i=s_init:-1:1 % count down since vector gets shorter
        if(abs(x(i))>num_sigma*sigma_l)
            x(i)=[];
        end
    end
    s_final=length(x);
    mu=mean(x); % calc new stats
    sigma_r=std(x);
    median_r=median(x);
    sigma_l=sigma_r/0.655;
    Pge=(s_init-s_final)/s_init;
    x_max=max(x);
    x=sort(x);
    delta=s_init-s_final;
    sum=sum+delta;
    if(delta<1)
        break
    end
    fprintf('\nIteration #%2i, after %2i outliers are removed,
\n', j, delta)
    fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f m, #GE
= %3i\n', mu, sigma_r, median_r, sum)
    j=j+1;
    sum;
end
fprintf('\n\n-----Final statistics-----
\n')
fprintf('mean = %4.3f m, sigma_rad = %4.3f m, median = %4.3f m, P_GE =
%4.3f\n\n', mu, sigma_r, median_r, sum/len_data)

%-----write residual file-----
----
s_out='GE_extract_output_file.txt';
fid=fopen(s_out, 'w');
fprintf(fid, '%5.2f\n', x);
fclose(fid);

diary off

```

I. GPS WEAPON ACCURACY CALCULATOR

```

%% Define Variable
sat_clock = .78;
ephemeris = .78;

```

```

tropos = .5;
ionos = 1.0;
noise = .4;
multipath=.5;
hdop = 1.3;
vdop= 2.0;

gdop = sqrt(hdop^2+vdop^2);

reciever = sqrt(ionos^2+tropos^2+noise^2+multipath^2);
space_clock = sqrt(sat_clock^2+ephemeris^2);

RSS_bias = sqrt(sat_clock^2+ephemeris^2+tropos^2+ionos^2);
RSS_rand = sqrt(noise^2+multipath^2);

UERE = sqrt(RSS_bias^2+RSS_rand^2);

%%Horizontal Error
nav_bias_h = (RSS_bias*hdop)/sqrt(2);
nav_rand_h= (RSS_rand*hdop)/sqrt(2);
nav_tot_h= sqrt(nav_bias_h^2+ nav_rand_h^2);

gc_bias_h =0;
gc_rand_h=.59;
gc_tot_h=sqrt(gc_bias_h^2+ gc_rand_h^2);

tle_bias_h = 0;
tle_rand_h=0;
tle_tot_h=sqrt(tle_bias_h^2+ tle_rand_h^2);

rss_bias_h = sqrt(nav_bias_h^2+ gc_bias_h^2+tle_bias_h^2);
rss_rand_h=sqrt(nav_rand_h^2+ gc_rand_h^2+tle_rand_h^2);
rss_tot_h= sqrt(rss_bias_h^2+rss_rand_h^2);

%%Vertical Error
nav_bias_v = (RSS_bias*vdop);
nav_rand_v= (RSS_rand*vdop);
nav_tot_v= sqrt(nav_bias_v^2+ nav_rand_v^2);

gc_bias_v =0;
gc_rand_v=.59;
gc_tot_v=sqrt(gc_bias_v^2+ gc_rand_v^2);

tle_bias_v = 0;
tle_rand_v=0;
tle_tot_v=sqrt(tle_bias_v^2+ tle_rand_v^2);

rss_bias_v = sqrt(nav_bias_v^2+ gc_bias_v^2+tle_bias_v^2);
rss_rand_v=sqrt(nav_rand_v^2+ gc_rand_v^2+tle_rand_v^2);
rss_tot_v= sqrt(rss_bias_v^2+rss_rand_v^2);

%Generate a random impact angle
sample = 10000;

```

```

average_impact = 78.5;
std_dev_impact = 3.88;
%impact_angle = randi([60 90], [1,sample]);
impact_angle = unifrnd(70,90,1,sample);
%sample =size(impact_angle);
%impact_angle = normrnd(average_impact,std_dev_impact,[1,sample]);

for i=1:sample
    angle = impact_angle(i);
    rep_bias = sqrt(rss_bias_h^2+(rss_bias_v/tan(deg2rad(angle)))^2);
    rep_rand = sqrt(rss_rand_h^2+(rss_rand_v/tan(deg2rad(angle)))^2);
    REP = sqrt(rep_bias^2+rep_rand^2)

    dep_bias = rss_bias_h;
    dep_rand = rss_rand_h;
    DEP = sqrt(rss_bias_h^2+rss_rand_h^2)
    sigma_x = REP/.6745;
    sigma_y=DEP/.6745;
    x(i) = 0+sigma_x*randn(1);
    y(i) = sigma_y*randn(1);
    r(i) = sqrt(x(i)^2+y(i)^2);
end
A=[x' y'];
R = r';

```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] M. Driels, *Weaponneering: Conventional Weapon System Effectiveness*, 2nd Ed. Reston, VA: AIAA, 2013.
- [2] R. Hogg et al., *Introduction to Mathematical Statistics*, 7th Ed. Upper Saddle River, NJ: Pearson, 2012.
- [3] M. Driels (private communication), 2013.
- [4] M. Driels, "Determining the significance of representing delivery accuracy by multiple distributions," unpublished manuscript.
- [5] O. Yakimenko, *Engineering Computations and Modeling in MATLAB/Simulink*. Reston, VA: AIAA, 2011.
- [6] *Global Positioning System Standard Positioning Service Performance Standard*. 4th Ed. Washington, DC: Department of Defense, 2008.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California